Predicting the behaviour of dislocation systems with machine learning methods

Henri Salmenjoki

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 22.11.2017

Supervisor

Prof. Mikko Alava

Advisor

D.Sc. (Tech.) Lasse Laurson

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Henri Salmenjoki

Title Predicting the behaviour of dislocation systems with machine learning methods

Degree programme Master's Programme in Engineering Physics

Major Engineering Physics **Code of major** SCI3056

Supervisor Prof. Mikko Alava

Advisor D.Sc. (Tech.) Lasse Laurson

Date 22.11.2017 **Number of pages** 49 **Language** English

Abstract

The irreversible changes in crystalline materials during plastic deformation are governed by the motion of dislocations – the line defects of the crystal structure. Experiments have shown that, on multiple scales, plastic deformation progresses through bursty events. On micron-scale, this is evident for instance in micro-pillar compression tests where the stress response of the sample exhibits strain bursts caused by dislocation avalanches. These avalanches have broad distributions of size and duration, and they cause the stress response to vary drastically from sample to sample resulting in fluctuations of sample properties.

Here the aim was to study if the initial dislocation configuration has effect on the stress response of the sample. The compression test was simulated with a two-dimensional discrete dislocation dynamics (2D DDD) model which captures the complex interactions of the dislocations. The initial systems were then characterized with a set of inputs that were fed to a regression neural network predicting the stress for certain strain. Although a perfect fit was not achieved, the network output and the stress values correlated especially with small strains in the start of the simulation. As an alternative approach, the dynamics of the dislocation system were taught to an interaction network, which is a modification of the basic neural network, to perform predictions of the entire simulations of creep tests.

Keywords Plastic deformation, dislocation dynamics, neural networks, interaction networks

Aalto University
School of Science

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Henri Salmenjoki

Työn nimi Dislokaatiosysteemien ominaisuuksien ennustaminen koneoppimismenetelmien avulla

Koulutusohjelma Master's Programme in Engineering Physics

Pääaine Engineering Physics **Pääaineen koodi** SCI3056

Työn valvoja Prof. Mikko Alava

Työn ohjaaja TkT Lasse Laurson

Päivämäärä 22.11.2017 **Sivumäärä** 49 **Kieli** Englanti

Tiivistelmä

Plastisen muodonmuutoksen aiheuttamat peruuttamattomat muutokset kiteisissä aineissa perustuvat dislokaatioiden eli viivamaisten hilavirheiden liikkeeseen aineen hilassa. Monet kokeet ovat osoittaneet, että plastinen deformaatio etenee purskeisesti – esimerkiksi mikropilarien puristuskokeissa tämä ilmenee venymäpurskeina voima-venymä-käyrässä. Venymäpurskeet syntyvät dislokaatiovyöryistä. Vyöryjen koko ja kesto vaihtelevat laajasti ja ne aiheuttavat sen, että voima-venymä-vaste on näytteiden välillä hyvin erilainen.

Tämän opinnäytetyön päämääränä olikin tutkia, onko näytteen alun dislokaatiorakenteella vaikutusta voima-venymä-vasteeseen. Data saatiin simuloimalla puristuskoetta kaksiulotteisen diskreetin dislokaatiodynamiikkamallin avulla. Puristuskoetta edeltäviä dislokaatiosysteemeitä kuvattiin parametreillä, jotka syötettiin neuroverkolle. Neuroverkon avulla sovitettiin epälineaarinen regressiomalli parametrien ja voima-venymä-vasteen välille. Vaikka sovitus ei onnistunut täydellisesti, neuroverkon ennustukset ja toteutuneet voima-venymä-arvot korreloivat erityisesti pienillä venymillä simulaation alussa. Vaihtoehtoisena lähestymistapana kokeiltiin myös vuorovaikutusverkkoa, joka on muunnos perinteisestä neuroverkosta. Vuorovaikutusverkolle opetettiin dislokaatiosysteemin dynamiikka, minkä avulla ennustettiin kokonaisia vakiovoimalla suoritettavia simulaatioprosesseja.

Avainsanat Plastinen deformaatio, dislokaatiodynamiikka, neuroverkot, vuorovaikutusverkot

Preface

The journey towards this thesis has been long and I could not have made it without the help of many, many people.

First of all, I owe a big thanks to professor Mikko Alava for the opportunity to work at the inspiring group Complex Systems and Materials of the Department of Applied Physics, Aalto University. For this thesis, I received superb guidance from Lasse Laurson, and also fruitful ideas from Arttu Lehtinen.

My beloved parents, Kimmo and Pirjo, have supported me throughout my life and I am grateful for their advice and lessons. Last but not least, I also want to thank my love, Tuulia, for doing the toughest work of keeping me sane during these years.

Otaniemi, 22.11.2017

Henri Joona Salmenjoki

Contents

Symbols and abbreviations

Symbols

σ	Stress
ε	Strain
$\mathbf{b}$	Burgers vector
$\mathbf{f}$	Force per unit length
L	System size
μ	Shear modulus
ν	Poisson ratio
χ	Dislocation mobility
ρ	Dislocation density
ρ_{GND}	Density of geometrically necessary dislocations
v	Dislocation velocity
f_{x1}	First Fourier coefficient of the ρ_{GND} field in x-direction
f_{y1}	First Fourier coefficient of the ρ_{GND} field in y-direction
$\varphi_j(\cdot)$	Activation function of neuron j
v_j	Input signal of neuron j
S	Score of a neural network

Abbreviations

SC	Simple cubic crystal
2D	two-dimensional
3D	three-dimensional
DDD	Discrete Dislocation Dynamics
ID	Initial deformation
PBC	Periodic boundary conditions
GND	Geometrically necessary dislocation
MLNN	Multi-layered neural network
IN	Interaction network

1 Introduction

Material failure can have catastrophic consequences. Average properties of materials can be deduced by experimenting on many samples, but the variance between the samples can rarely be accounted for beforehand which can lead to surprising failure or dysfunctional behaviour. Therefore, predicting for instance the toughness of individual samples, among other material properties, is a pursued goal in materials science.

A central aspect of material toughness is the way materials deform plastically. Plastic deformation comprises the irreversible changes in the material structure caused by external forces. It is fundamentally governed by motion of dislocations – the topological defects of the material structure. External forces move the line defects which causes the sample to strain and eventually yield. What makes the plastic deformation complex is its unpredictable nature – for example acoustic emission measurements during single crystal ice creep experiments and compression tests of micro-pillars have shown that on many scales, the plastic deformation progresses through bursty events [1, 2, 3]. The bursty deformation appears as a random process as the sizes, durations and waiting times of the bursts show broad distributions. However, the connection between the dislocation structures and the bursts has not been thoroughly studied.

Meanwhile, machine learning methods have become more and more popular in physics research [4]. This is only natural as machine learning provides efficient tools for analysis of large datasets. Especially neural networks have proven their usefulness as they are capable of pattern recognition and non-linear input-output mapping [5]. Moreover, neural networks have inspired many induced models, one being the interaction network which has shown promising results for learning the underlying dynamics of a physical system [6]. These kinds of developing tools provide a new approach to analysis of complex physical phenomena.

The aim in this thesis is to apply machine learning methods to study the bursty plastic deformation process. Here, this reduces to using neural networks to observe if the stress response of a system can be predicted from the initial dislocation structure. The data for study is obtained with a two-dimensional model of discrete dislocation dynamics and the model is implemented with a procedure that tries to imitate the micro-pillar compression experiment. Additionally, the interaction network is applied to constant stress loading simulations of the same model to deduce if it is able to learn the dynamics of the dislocations. This way, the ability of the interaction network to function on data from a highly complex system is tested and discussed.

The thesis is sectioned as follows. Firstly, Section 2 provides the key concepts of plastic deformation and dislocation motion along with a summary of recent studies of dislocations and machine learning methods in physics. Then, Section 3 presents the used research methods, namely the discrete dislocation dynamics model of the

simulations and the neural and interaction network implementations for the stress response prediction and interaction learning. After that, Section 4 discusses the results of the predictions from the networks. Also the feature extraction required to characterize the initial dislocation structures is considered here. Section 5 then addresses the questions arising from the results. And finally, Section 6 summarises the findings.

2 Background

This Section describes the motivation behind this thesis. First, Section 2.1 gives an introduction to crystal dislocations and their concepts. Then, Section 2.2 discusses the recent studies of dislocation structure and machine learning methods in materials science.

2.1 Crystal dislocations

In materials science, a significant emphasis is put on the response of the material to external forces. Basically, this means studying the stress-strain relation: How much of deformation does certain stress cause. When focusing on ductile materials, such as most alloys, the stress-strain response contains two distinct regions, the *elastic* and *plastic deformation*. This is depicted by Figure 1 which shows the stress-strain curve of a typical ductile material. The elastic part of the curve is linear, and it

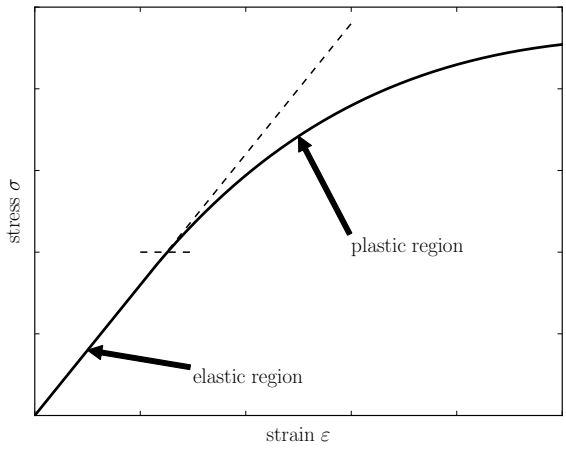


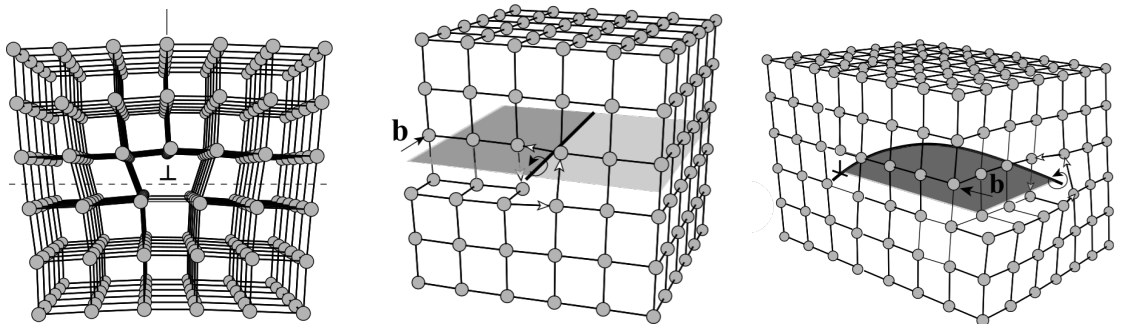
Figure 1: Stress-strain curve of a ductile material. Elastic and plastic regions are observable and they are separated by the short horizontal dashed line. This is also called the *yield point* of the material.

corresponds to reversible deformation. In other words, removing the external stress in the elastic region causes the material to recover to its initial state. On the other hand, the plastic deformation is the opposite – changes in the material structure are permanent and the properties of the material shift.

Originally, plastic deformation was observed in metals as slip bands, which mean parts of material shearing with respect to rest of the body [7]. Although these observations were made in the end of the nineteenth century, explanation for the plastic behaviour required closer inspection of the material. Eventually, the

development of x-rays provided the right tool to confirm crystalline structure of metals and, moreover, deduce that real life crystals are imperfect. In other words, crystalline materials contain defects that are irregularities in the crystal structure. From the possible irregularities, line defects arising from slightly slid atomic planes, that is *dislocations*, proved to be responsible for the plastic deformation [7, 8].

To obtain a clearer perception of dislocations in crystals, Figure 2a shows an *edge dislocation* in a simple cubic (SC) crystal system. SC crystal denotes a lattice structure where atoms are positioned in the corners of a unit cube with a lattice vector **a**, and the cube is duplicated in every direction indefinitely to form the material bulk. Then, the edge dislocation appears as a partial extra atom layer introduced to the structure. The '⊥' marker shows the position of the dislocation line which



(a) (b) (c)

Figure 2: Illustration of (a) an edge dislocation, (b) a screw dislocation and (c) a mixed dislocation in a SC crystal. Figures extracted from [9].

is perpendicular to this page. Consequently, applying external stress to the crystal causes the dislocation to move.

Options for the movement are *conservative* and *non-conservative* [9]. In conservative motion, the dislocation travels along its glide-plane. In Figure 2a, this means the plane marked with the dashed line, and Figure 3 shows the motion when a shearing force T_x is applied to the top surface of the material while fixing the bottom surface. The dislocation jumps from one atomic layer to another by breaking atomic bonds at the dislocation core and forming new bonds behind the dislocation, until it reaches the edge of the material, where it has caused a displacement of **b** between the upper and lower parts of the crystal. On the other hand, non-conservative motion covers for dislocation motion perpendicular to the glide plane, which is often called dislocation climb. In practice, this requires diffusion of atoms as in the case of Figure 3 either more atoms need to be added or some atoms need to be removed for the dislocation to be able to move in y-direction. On the other hand, edge dislocations are not the only type of dislocations. Figure 2b shows a *screw dislocation* which forms when two parts of the crystal are torn with respect to each other. Moreover, a dislocation can be a mixture of edge and screw, as shown in Figure 2c.

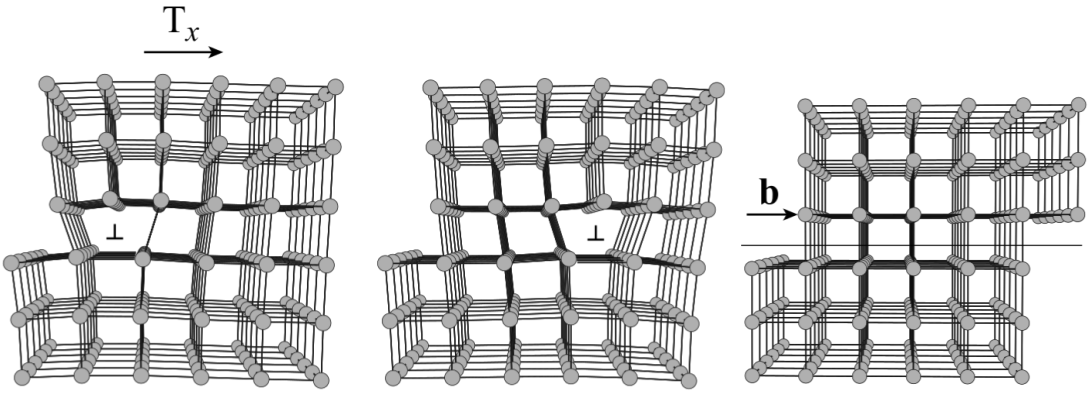


Figure 3: Consecutive pictures of edge dislocation movement in a SC crystal under a shearing force T_x. Figures extracted from [9].

In general, any dislocation can be associated with the Burgers vector **b**, the 'topological charge' of the dislocation [8, 9]. The Burgers vector can be deduced by surrounding the dislocation line with a Burgers circuit as depicted by Figure 4. In practice, the Burgers vector is the amount of difference between Burgers circuits in the test crystal and a perfect crystal. As the figure shows, the sign of the Burgers vector is arbitrary in the sense that the direction of circulation of the Burgers circuit, namely line sense, defines it. Consequently, a reversed dislocation has an opposite Burgers vector and defining the Burgers circuit around both dislocations leads to zero Burgers vector.

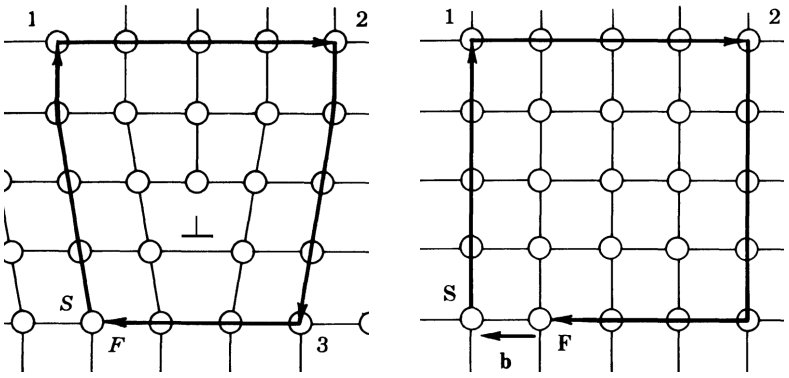


Figure 4: Burgers circuit $S \rightarrow F$ illustrated (a) in crystal with an edge dislocation and (b) in corresponding perfect crystal. The mismatch between the circuits is the Burgers vector **b**. Figure modified from [7].

With the definition of the Burgers vector, force acting on a dislocation can be computed. Presuming the situation of Figure 3a where the force T_x acts on a crystal

of size $L_x \times L_y \times L_z$, the stress is obtained as

$$\sigma_{xy} = \frac{T_x}{L_x L_z}. \tag{1}$$

Now if the dislocation line sense is chosen so that $\mathbf{b} = [b_x\ 0\ 0]$, the work W done by the force, when the top surface has moved by $\mathbf{b}$ (Figure 3c), is $W = b_x T_x$. On the other hand, the same work is done to the dislocation by a driving force f_x that is defined per unit length. Therefore, moving a dislocation of length L_z from the left edge to the right edge of the crystal requires work $W = L_x f_x L_z$. Combining these gives f_x the form

$$f_x = b_x \sigma_{xy}. \tag{2}$$

Generalizing this to three dimensions yields the Peach-Koehler formula,

$$\mathbf{f} = (\sigma \cdot \mathbf{b}) \times \xi, \tag{3}$$

which gives the force per unit length $\mathbf{f}$ for dislocation with Burgers vector $\mathbf{b}$ and line direction ξ in a stress field described by the tensor σ.

The stress field is naturally affected by any external forces, but also by other dislocations. As a dislocation is an impurity in the crystal, it generates a local stress field of its own. For an edge dislocation in an infinite medium, the stress tensor is, expressed in the polar coordinates where z direction coincides with the dislocation line [7],

$$\begin{aligned} \sigma_{rr} = \sigma_{\theta\theta} &= -\tfrac{\mu b \sin\theta}{2\pi(1-\nu)r} \\ \sigma_{r\theta} &= \tfrac{\mu b \cos\theta}{2\pi(1-\nu)r} \\ \sigma_{zz} &= -\tfrac{\mu b \nu \sin\theta}{\pi(1-\nu)r} \\ \sigma_{rz} = \sigma_{\theta z} &= 0. \end{aligned} \tag{4}$$

Here μ denotes the shear modulus and ν the Poisson ratio. Interestingly, the stress field of an edge dislocation is anisotropic and the direction is also affected by the sign of the Burgers vector. Moreover, the field decays $1/r$ so it has long range [7].

As can be noted from the movement of the dislocation in its glide plane (Figure 3), the movement is restricted by a periodic potential that represents the repeating barrier of breaking atomic bonds of the next atomic layer [9]. On the other hand, the potential can be influenced by other crystal defects such as atomic vacancies or precipitates. The combination of these effects can be employed to form a mobility function $\mathbf{M}(\cdot)$ that connects the dislocation velocity $\mathbf{v}$ given the force per unit length, namely

$$\mathbf{v} = \mathbf{M}(\mathbf{f}). \tag{5}$$

The mobility function is fundamental to any dislocation simulation as it acts as the representation of the real world material which is being simulated.

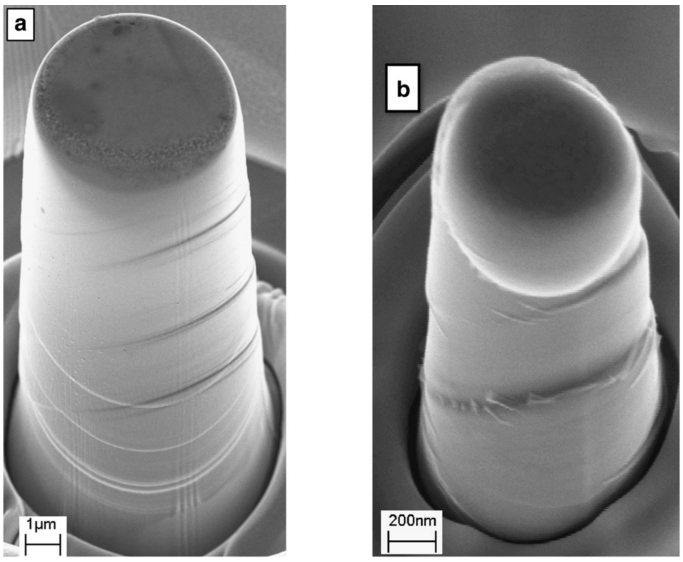


Figure 5: Pictures of two aluminium micro-pillars after compression, with initial height (a) 6.8μm and (b) 0.8μm. Image retrieved from [3].

2.2 Recent studies

Crystal plasticity

Intriguing feature of crystal plasticity is its crackling nature: plastic deformation is generated by bursts of activity instead of smooth and uniform motion [1]. In the atomic-scale this makes sense due to the discrete steps that the dislocations take, but deformation has shown intermittent behaviour also in the micro- and macroscopic scales. This appears in different forms: loading experiments with micron-scale samples have produced jerky stress-response while acoustic emission measurements of loaded macroscopic samples of ice have displayed bursty activity as well [2, 3]. Moreover, the size-scale itself affects the plasticity as samples in the micron- and nano-scales have shown that the yield stress is larger the smaller the sample [10, 11]. This phenomenon is also known as the *extrinsic size effect*.

The burst-like deformation is a feature observed in experiments and simulations. Nowadays, as materials science applications pursue smaller and smaller scales, properties of micro-pillars are widely studied. Compression tests of pillars are a common procedure where jerky deformation is encountered [3, 12]. Figure 5 shows a pair of aluminium micro-pillars after such compression experiment. The stress-strain curves of the corresponding samples are presented in Figure 6. The origin of the bursts is still partly unknown but, according to the most common knowledge, it lies in the dislocation motion – the crystal dislocations move in avalanches [1, 13, 14, 15, 16]. These avalanches cause the deformation process to advance in a random-like manner, as the size s and duration T of the avalanches fluctuate. In fact, statistics of the avalanches underline the large variation of their sizes. The size of the bursts has been

indicated to follow power-law distribution $P(s) \sim s^{-\tau}$, although a possible cut-off size could limit the maximum bursts due to strain-hardening effects [17, 18, 1].

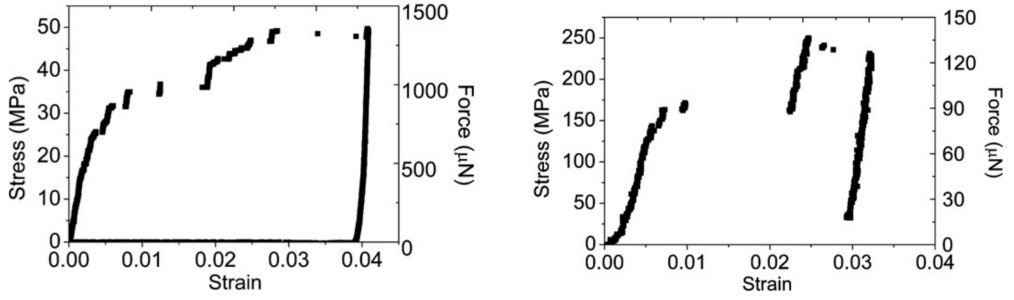


Figure 6: Stress-strain curves of the samples in Figure 5. Images from [3].

The bursty deformation provokes also the question about the yielding of the micron-scale samples: at which point can the sample be considered to have yielded? Experiments and simulations have indicated that averaging the stress-strain behaviour of many similar samples provides stress-strain curves where a threshold for shift in the response, that is a point similar to yielding, can be defined [12]. On the other hand, single sample yielding is considered as a purely random case due to the assumed stochastic nature of the bursts that define the sample stress response [19]. This has inspired the development of some stochastic models for dislocation simulations (for instance [19]). However, commonly used dislocation dynamics models, that replicate avalanche behaviour observed in experiments, neglect any stochastic effects and are thus purely deterministic – but extremely complicated.

Machine learning methods in physics

Due to the huge improvements in computing power, machine learning is a hot topic in many branches of science and technology. In physics, many methods, especially *artificial neural networks*, have proven their ability as a tool for regression and sample identification. Indeed, applying machine learning with the vast amount of data produced by experiments or simulations provides a plethora of new possibilities.

Lately, *convolutional neural networks*, that are used for pattern recognition by the algorithms of large technology companies, have assisted in observing phase transition of some statistical models, such as the Ising model [20, 21, 22]. The idea of *interaction networks*, that learn the underlying physics of presented configurations, help study complex systems in a new way [6]. Also materials science has utilized machine learning [4]. For instance *clustering* samples by their diffraction spectrum or obtaining interatomic potentials with neural networks trained with data from density functional theory show that machine learning techniques can provide tools to reduce the costs of expensive computations [4, 23].

Moreover, machine learning enables the predictions of single sample properties. Recently, the output from digital image correlation, which measures the local deformation of a sample, was studied with *principal component analysis* [24]. This way, the samples could be distinguished to groups by their shear history. Another intriguing concept is the modelling of the stress-strain relation of a material with neural networks, which has invoked research concerning metals during hot deformation [25, 26]. On the other hand, direct study of the connection between the dislocation structure and the material toughness has not yet been conducted. Although some model fitting with dislocation cell structure as parameters was attempted in [27], testing of the model was omitted.

The aim of this thesis is to analyse the possibility of predicting the stress-strain curve from the initial dislocation structure, and this way to study the presumed stochastic nature of the bursty deformation. Hence, the simulation of complex dislocation configurations with a two-dimensional model is combined with the fitting capabilities of neural networks [5, 28]. First approach is to find a description of the initial system that, when used as input to a regression network, provides information of the stress response of the sample. Secondly, the interaction network is applied to learn the dislocation dynamics, and to study its properties.

3 Research methods

In this Section, the used research methods are discussed. First, Section 3.1 describes the simulation methods that produced the data for analysis. Then, the operating principles of the applied machine learning techniques, namely artificial neural networks, are described in Section 3.2. Finally, Section 3.3 discusses the implementation of the networks; first the neural networks for the regression problem, and then the interaction network for the interaction predictions.

3.1 Discrete Dislocation Dynamics simulations

Two-dimensional (2D) discrete dislocation dynamics (DDD) simulations are a common procedure in dislocation research as they are easy to implement and light to compute but still exhibit complex behaviour [12, 28, 29]. The idea in 2D DDD simulations is to simulate the motion of parallel edge dislocations by focusing on a cross-section of a crystal that is perpendicular to the straight dislocation lines. This way, the dislocations act as interacting points on the cross-section plane. Furthermore, the motion is restricted to the so-called single-slip, which means that the dislocations are forced to move along their original glide plane and climbing is neglected.

Illustration of an example simulation configuration is presented in Figure 7. The dislocation lines are positioned parallel to the hidden z-axis and their Burgers vectors,

$$\mathbf{b} = \pm b\mathbf{u}_x = sb\mathbf{u}_x, \tag{6}$$

are parallel to the x-axis which is also their glide plane. First half of the the dislocations are initialized with positive and the other half with negative Burgers vectors, and the direction is represented with the parameter $s = \pm 1$. In the Figure, these are represented with red, upward pointing and blue, downward pointing symbols, respectively. During the simulation, dislocations with opposite sign, that drift closer to each other than the cut-off distance b (that is, the magnitude of their Burgers vector), are annihilated. On the other hand, creation of dislocations is neglected.

The dislocation–dislocation -interaction originates from the shear stress field σ_d generated by an edge dislocation [7],

$$\sigma_d(\mathbf{r}) = \sigma_d(x, y) = \frac{\mu b}{2\pi(1-\nu)} \frac{x(x^2 - y^2)}{(x^2 + y^2)^2}. \tag{7}$$

Here, the coordinates x and y are the distances from the dislocation in the respective directions. As there is no dislocation climb, the dislocations move only in the x-direction. As the dislocation induced stress field of Equation (7) is a long-range effect due to the $\sim 1/r$ decaying, the simulation region is implemented with periodic boundary conditions (PBC). This way, the long-range forces imitate the forces found

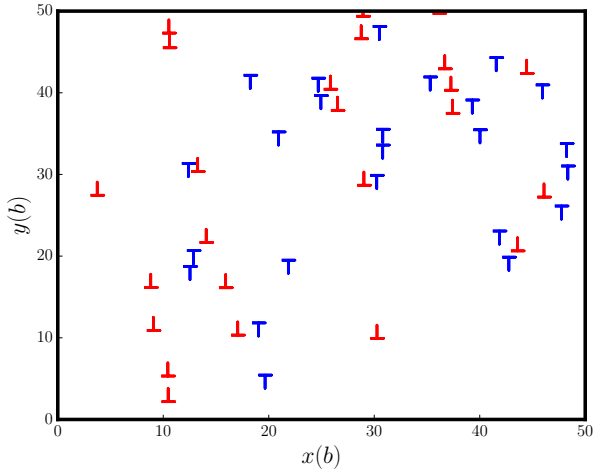


Figure 7: Illustrative plot of the dislocations inside the simulation box before the start of loading. Dislocations with positive and negative Burgers vectors plotted with red, upward pointing markers, and blue, downward pointing markers, respectively. The distances are measured with respect to the magnitude of the Burgers vector of the dislocations, b.

in the bulk. In the stress field calculation, this is accounted for by computing also dislocations from infinite images of the simulation box in the x-direction. Thus, the corrected stress field becomes

$$\sigma_d(x,y) = \frac{\mu b}{2\pi(1-\nu)} \sum_{n=-\infty}^{\infty} \frac{(x+nL)\left[(x+nL)^2 - y^2\right]}{\left[(x+nL)^2 + y^2\right]^2}, \tag{8}$$

where L is simply the size of the simulation box. Using residue theorems, the sum obtains the exact form [15, 30]

$$\sigma_d(x,y) = \frac{\mu b}{2(1-\nu)} \frac{1}{L} \frac{\sin\left(\frac{2\pi x}{L}\right)\left[\cosh\left(\frac{2\pi y}{L}\right) - \cos\left(\frac{2\pi x}{L}\right) - \frac{2\pi y}{L}\sinh\left(\frac{2\pi y}{L}\right)\right]}{\left[\cosh\left(\frac{2\pi y}{L}\right) - \cos\left(\frac{2\pi x}{L}\right)\right]^2}. \tag{9}$$

Combining Equation (9) to the assumption that the dislocation motion is overdamped, the equation of motion for the ith dislocation is [12, 29]

$$\frac{1}{\chi b}\frac{\mathrm{d}x_i}{\mathrm{d}t} = s_i b \left(\sum_{j \neq i} s_j \sigma_d(\mathbf{r_j} - \mathbf{r_i}) + \sigma_{ext} \right), \tag{10}$$

where the sum is taken over all but the ith dislocation, χ is the dislocation mobility, s_i and s_j denote the signs of the Burgers vectors of dislocations i and j, and σ_{ext} is the applied external stress. While solving Equation (10), the computational units are chosen so that all the distances are measured in units of the Burgers vector length b, time is measured in units of $2\pi(1-\nu)/\mu\chi b$ and stress in units of $\mu/2\pi(1-\nu)$.

Code implementation for the 2D simulation is the same as used by Ovaska et al. in [29]. The code uses the fifth order Runge-Kutta with adaptive timestep to solve Equation (10) for every dislocation. Data for the stress-strain predictions was obtained with two different methods. First, the basic simulation procedure is as follows: In the beginning, N dislocations are initialized to totally random positions inside the simulation box. Then, to ensure a (somewhat) stable state, the system is given time to relax with $\sigma_{ext} = 0$. When the relaxation is complete, the loading procedure is started. The simulations for the regression of the stress response used the *quasistatic stress ramp*, which tries to mimic slow compression experiments. During the ramp, the external stress is increased with a rate $\dot{\sigma}_{ext} = 10^{-5}$ during the simulation. However, once the average velocity of the dislocations exceeds a pre-defined threshold $v_{th} = 3 \cdot 10^{-4}$, the stress ramp is stopped until the motion ceases. These events are called avalanches and they are mostly responsible for the accumulated strain ε. This can be seen in Figure 8, where example stress–strain curves of five different simulations are plotted. The two phases, stress ramps and avalanches, are distinguishable as the parts where the stress and strain increase and the parts where only strain increases, respectively [13].

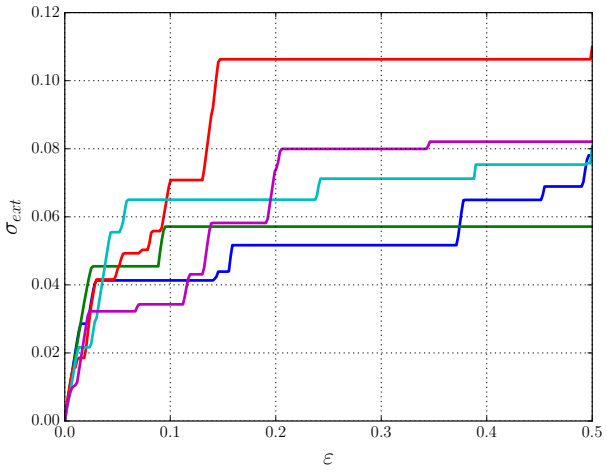
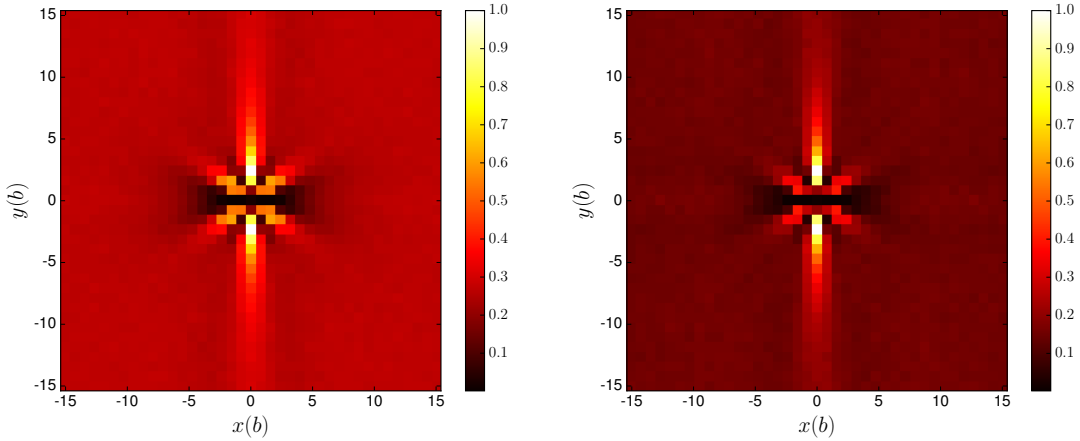


Figure 8: Stress–strain curves of five random systems driven with the basic routine. The stress ramps, where the external stress is increased, and the strain avalanches, where the dislocations move with constant external stress, are seen in the figure.

As an alternative for the basic simulation routine, system initialization with initial deformation (ID) was also used. This means that the initially relaxed system is first driven quasistatically until it reaches strain $\varepsilon_{\mathrm{ID}} = 0.2$ and then immediately relaxed with $\sigma_{\mathrm{ext}} = 0$. After this second relaxation, the quasistatic stress ramp is executed as with the basic routine. The idea with initial deformation is to find more stable states than with the basic routine as the dislocations form physically relevant structures, such as dislocation dipoles, instead of the totally random initial positions. For this thesis, about 8700 systems were calculated with the basic routine and correspondingly about 1700 with the ID routine. To show the difference in



(a) (b)

Figure 9: Dislocation-dislocation correlation function $d(x, y)$ before the loading for systems with (a) basic routine and (b) ID.

the dislocation configurations between the routines, Figure 9 shows the dislocation-dislocation correlation functions $d(x, y)$ calculated for the two groups. Here, a modified definition of $d(x, y)$ is used, namely [15, 31]

$$d(x,y) = \frac{\sum_{\text{system}} \sum_{i=1}^{N} \rho_i(x,y)}{\max_{x,y}\left[\sum_{\text{system}} \sum_{i=1}^{N} \rho_i(x,y)\right]}, \tag{11}$$

where the first sum is taken over all of the simulated systems and the second sum over all the dislocations in the system, and $\rho_i(x, y)$ denotes the dislocation density at (x, y) relative to dislocation i. In other words, the functions are obtained simply by summing neighbouring dislocations of any sign at relative positions with respect to a test dislocation, and this is repeated for every dislocation of every system. To get comparable results for the two groups with different amount of data, the functions are scaled with their maximum values. Figure 9a, which shows the correlation function for basic loading routine data, implies that, although there are dislocation wall structures and dipoles, also random structures are present. Especially when comparing to $d(x, y)$ of ID routine seen in Figure 9b, where the correlation is more strongly peaked to closer the dislocation. Another factor causing the structure difference is that the initial loading causes some dislocation annihilations that decrease the dislocation density of the ID systems relative to the control systems. This also leads to ID systems being stronger than the basic systems. This can be seen in Figure 10 where the average stress-strain curve is plotted for the compared groups.

However, with both of the procedures, there is a problem arising from the PBC. During the avalanches, single dislocations can travel even longer distances than the box width. Moreover, at some strain in the simulation, the dislocations may form a single wall that moves uniformly in the simulation box, and the structure starts to circulate over and over, passing the periodic boundaries once the stress achieves

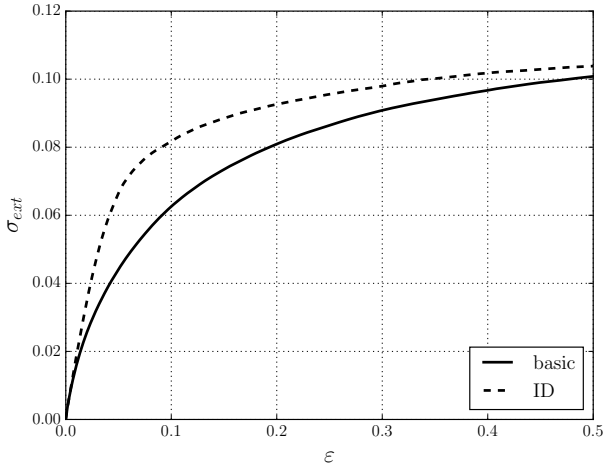


Figure 10: The average stress-strain curve for the basic systems (solid line) and ID systems (dashed line). For any simulation strain ε, ID systems need on average larger stress σ_{ext}.

a large enough value. This wall formation is illustrated in Figure 11. These facts introduce a problem of model validity: Can the system be realistic in the physical sense when a dislocation 'returns' to its original position by circulating the entire simulation box?

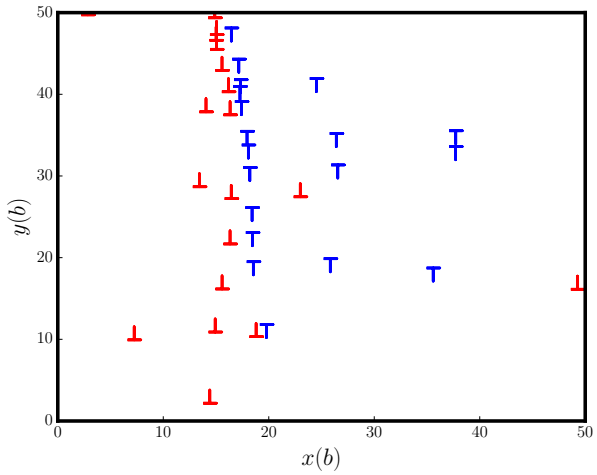


Figure 11: The same system as seen in Figure 7, but now at the end of the loading ($\varepsilon \approx 0.5$). The dislocations have formed a wall which moves almost uniformly in an endless avalanche. Sometimes, this is classified as the point of yielding (for instance [12]) but here, systems driven this far are left out of considerations.

In the simulations for this thesis, the problem of validity was even more emphasized. To be able to do predictions, a substantial dataset is required and this is easiest to

achieve by simulating a small system. For the dataset to be generated, the system box width is set to $L = 50$ and the number of generated dislocations to $N = 100$, though the dislocation count decreases to $N \approx 60$ during the relaxation with the basic routine, and $N \approx 50$ with ID routine; these are notably smaller than $L = 200$ and $N = 1600$ that were used by Ovaska et al. in [29], but the dislocation density is similar in the initialized systems. In small dislocation systems, the problematic box circulation is more common as there are less dislocations to block the movement. Therefore, to ensure the validity of the simulations, the simplest solution is to focus on the strains where the dislocations have not experienced full circulation. Figure 12 shows the proportion of driven systems, where at least one dislocation has travelled a distance larger than L, as a function of strain. The figure shows that, especially

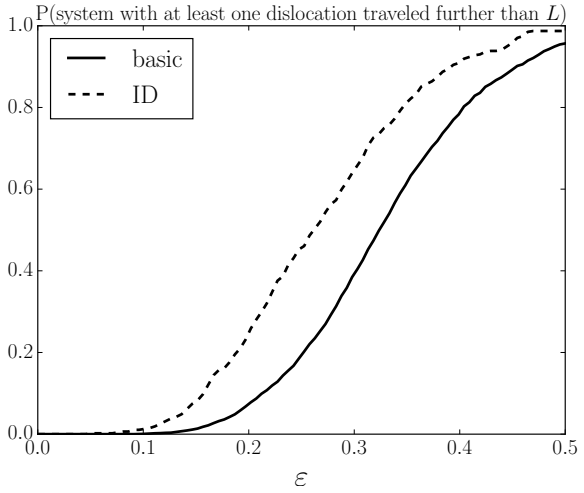


Figure 12: Proportion of the systems, where at least one dislocation has moved a greater distance than the simulation box width L, as a function of strain for the two different simulation procedures.

among the initially deformed systems, at least one dislocation tends to return to its original position already before the material 'flows'. Dislocations circulating sooner in the ID systems is a natural consequence of smaller dislocation density compared to the system driven with the basic routine. Also in the end of the simulation, virtually every system with both simulation procedures has experienced full circulation of a dislocation. Therefore, the predicting is concentrated on small strains, $\varepsilon \leq 0.2$, where only minority of the systems have, in a sense, lost their physical validity.

3.2 Neural networks

3.2.1 Structure of a neural network

The basic building piece of an artificial neural network is the *perceptron*. It was introduced as an computational concept already in the 1950s [5, 32]. The perceptron is illustrated in Figure 13. The idea of the perceptron is to mimic the behaviour

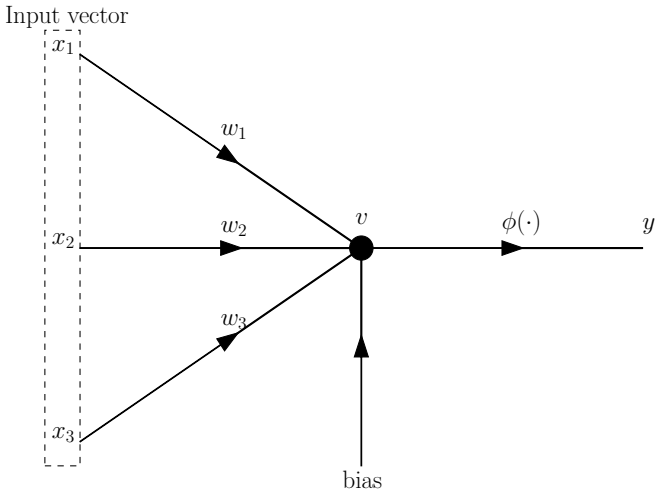


Figure 13: An illustration of a perceptron with three input connections. The values of the input vector $\mathbf{x}$ are multiplied by the connection weights and summed with the bias. The obtained sum is then passed to the activation function φ which produces the perceptron output y.

of a biological neuron of human brain. The input connections, denoted with their corresponding weights w_i in the figure, act as synapses in the biological system, where they pass signals between the neurons. Inside the perceptron, the m incoming signals x_i are multiplied with the corresponding weights and summed to an external bias b. Thus, the signal v is the linear combination of the inputs, and with definitions $\mathbf{x} = [1, x_1, x_2, \ldots, x_m]$ and $\mathbf{w} = [b, w_1, w_2, \ldots, w_m]$ it can be written as

$$v = \mathbf{w}^T\mathbf{x} = \sum_{i=0}^{m} w_i x_i. \tag{12}$$

Finally, to obtain the output y of the perceptron, signal v goes through the activation function $\varphi(\cdot)$ assigned to the perceptron.

Choosing the activation function depends on the object of the network. In fact, if a single perceptron is implemented with a limiter φ_{limiter}, that yields -1 in case of

negative v and $+1$ otherwise,

$$\varphi_{\text{limiter}}(v) = \begin{cases} -1 & v < 0 \\ +1 & \text{otherwise} \end{cases} , \tag{13}$$

the perceptron is already able to classify elements between two groups that are separated by a hyperplane in the space spanned by the input parameters x_i. This single perceptron is basically the simplest architecture for a neural network. It can also be modified to linear regression by using linear activation function $\varphi_{linear}(v) = v$, or extended to classify many groups by adding parallel perceptrons. However, the groups still can not be intertwined in classification, as single layer networks are not able to learn any non-linear correlations.

As the case here is to study complex dislocation systems, where the relations are supposed to be non-linear, single perceptrons are not sufficient. Hence the use of *multi-layered neural networks* (MLNN). Figure 14 shows an example MLNN with three layers: Input, hidden and output. By combining layers of perceptrons so

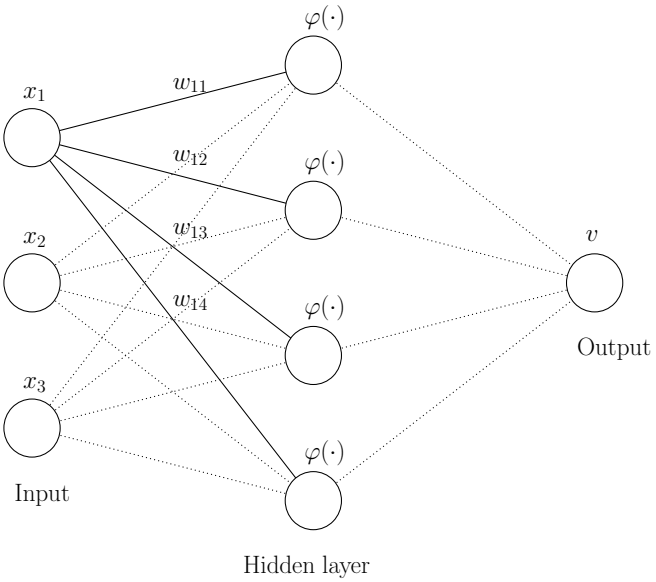


Figure 14: An illustration of a MLNN with three layers. The input values $x_{1...3}$ are passed to the hidden layer, from where the calculated signals continue to the output layer. The network is dense, that is all the neurons in the previous layer are connected to all of the neurons of the next layer. All the connections are assigned with a weight; the weights for the connections leaving from the first input neuron to the hidden layer are shown in the figure.

that there is one (or more) hidden layers between the input and output signal, the learning ability of the network increases. In fact, assigning only one hidden layer with an activation function fulfilling certain conditions enables the network to learn a uniform, non-linear approximation from a set of inputs x_i to a set of outputs y_i.

This is ensured by the universal approximation theorem. The theorem and its proof can be found for instance in [33] as they are omitted here.

Now, the existence of an approximation is ensured with a set of network connection weights, but only for non-constant, bounded and monotone-increasing continuous activation functions [5]. Thus, the decision of the activation function is crucial and different functions have been tested in neural network construction. The most commonly used options are the sigmoid function

$$\varphi_{\text{sigmoid}}(v) = \frac{1}{1+\exp(-v)}, \tag{14}$$

and the hyperbolic tangent

$$\varphi_{\tanh}(v) = \tanh(v). \tag{15}$$

Additionally the rectifier,

$$\varphi_{\text{rectifier}}(v) = \max(0, v), \tag{16}$$

although being unbounded, can be used. Actually from these functions, the rectifier has shown better convergence during training and, therefore, it is more popular choice especially in more complex networks, such as convolutional neural networks (CNN) that classify patterns in for instance images [34, 35]. Finally as another note, the existence of the approximation does not contain any information about the optimal network structure: although only one hidden layer is needed to construct the approximation, more hidden layers can achieve better generalization to data that is not used in the approximation construction, that is the training of the network. Moreover, the optimal number of units in the hidden layers is basically arbitrary, and usually finding the best layer sizes is an effort of trial and error [5, 36].

3.2.2 Training the network

The training of a neural network aims to find the connection weights, that minimize the *total error energy* ξ produced by the network. Provided the input vector $\mathbf{x}(n)$, the error $e_j(n)$ at neuron j between the desired output $d_j(n)$ in the output layer and the output signal $y_j(n)$ is written

$$e_j(n) = d_j(n) - y_j(n). \tag{17}$$

Then, the total error energy is

$$\xi(n) = \frac{1}{2}\sum_{j=1}^{n} e_j(n)^2 \tag{18}$$

where the sum is taken over all neurons in the output layer. As the goal of this thesis is to do regression on a single output variable, the output layer contains only one neuron and the error energy simplifies to $\xi(n) = \frac{1}{2}e_1(n)^2$ for the networks used in this thesis [5].

The actual training can be implemented with the *batch* or *on-line learning method*. In batch learning, the entire training data is passed to the network and the weights are updated according to the average error energy of the samples. On the other hand in on-line learning, one training epoch consists of feeding the training set to the network one sample at a time and the weights are updated after every sample. For the networks of this thesis, the latter method is used. This enables the use of the *back-propagation algorithm* to compute the weight updates. It is a popular modification of the gradient descent where the error of the output layer is distributed to the previous layers to be able to update connection weights.

The derivation of the algorithm can be found for instance in [5]. Briefly, the weight updates of connections to neuron j are obtained by calculating the gradient step to reduce the error energy,

$$\Delta w_{ji} = -\eta \frac{\partial \xi(n)}{\partial w_{ji}}. \tag{19}$$

In the above equation, η is called *the learning rate* and its correspondence in the gradient descent is the step size and in on-line learning η has a crucial impact on the training convergence. The partial derivative $\frac{\partial \xi(n)}{\partial w_{ji}}$ can be calculated with the chain-rule so that the weight updates can be explicitly computed from

$$\Delta w_{ji} = \eta \delta_j(n) y_i(n), \tag{20}$$

where

$$\delta_j(n) = \begin{cases} e_j(n)\varphi_j'(v_j(n)), & j \text{ in output layer} \\ \varphi_j'(v_j(n)) \sum_k \varphi_k'(v_k(n)) w_{kj}, & j \text{ in hidden layer, } k \text{ in next layer} \end{cases}. \tag{21}$$

Therefore, the update is dependent on the input from the previous layer y_i, derivative of the activation function at the neuron $\varphi_j'(v_j(n))$ and either the error $e_j(n)$, if neuron is in the output layer, or the derivatives of the activation functions and the connection weights of the next layer, if neuron is in a hidden layer. One training step contains two phases, the forward and the backward pass . First, the output is calculated for certain input vector by passing the signal towards the output layer. Then the error is passed from the output layer towards the input layer while updating the connection weights according to Equation (20) [5].

There are also multiple tweaks to the classic back-propagation algorithm to improve its convergence. For the networks of this thesis, the algorithm is implemented with the momentum updates, where the previous weight updates at step $t-1$ of the training affect the next update at step t, namely [36]

$$\Delta w_{ji}^t = -\eta \frac{\partial \xi(n)}{\partial w_{ji}} + \alpha \Delta w_{ji}^{t-1}. \tag{22}$$

Here $\alpha \in [0.5, 1]$ is the momentum parameter. Using the previous updates smooths the convergence of the training and it helps the algorithm to avoid irrelevant local minima [5].

To measure the the convergence of the network, that is deciding when to stop the training, *early stopping* is a common procedure. This means dividing the data set to three parts: training, validation and test sets. The first one is naturally used to calculate the weight updates. On the other hand, the validation set is a smaller set that measures the error at every epoch. This way, the training can be stopped when the network starts to overfit, which means that the error of the validation starts to increase while the training error continues decreasing. In other words, this is the point when the fitted network generalises in the best manner. Finally, the test set is then used to compute the actual score S of the network with a set excluded from the training phase. S is measured with equation

$$S = 1 - \frac{\sum_i (d_i - y_i)^2}{\sum_i (d_i - \langle d \rangle_i)^2}, \tag{23}$$

where d_i is the desired output to input x_i, y_i the corresponding network output and $\langle d \rangle_i$ mean of the desired outputs. Hence, the desirable outcome of perfect fit means $S = 1$ as $d_i = y_i$, and the fit can be infinitely bad ($S = -\infty$).

3.3 Network implementation

The technical details of the used networks are presented here. All of the networks were implemented with the open source software library TensorFlow for Python [37].

3.3.1 Regression network

The regression network, which tries to predict the quasistatic stress-strain response from the initial state, used the simple network structure similar to Figure 14: features of the initial configuration form the input vector and the output is the stress required to produce strain ε_0. Therefore, to predict the stress for various strains, a network for every strain is trained. All these networks share the same structure. In the testing phase, the results for a network with one hidden layer with 300 neurons (which achieved better results than smaller hidden layers) provided similar results as a network with three hidden layers containing 20 neurons each. Hence, to follow the practice of leaner and deeper over wider and shallower, the latter structure was used.

The MLNN training parameters, learning rate η and momentum α, as well as the initial connection weights w_{ij}^0 were drawn from uniform distributions of $n \in [5 \cdot 10^{-5}, 10^{-3}]$, $\alpha \in [0.5, 0.95]$ and normal distribution $w_{ij}^0 \sim \mathcal{N}(0,\, 0.01)$, respectively, and the networks were trained with multiple random seeds. The networks were trained with the early-stopping criterion and, finally, the networks with the best validation score were chosen as the predictors. The training set for the predictions concerning the basic routine consisted of 6000 samples, while for the systems driven with ID, the training set had 1500 samples. Correspondingly,

both the validation and test sets had 600 data points with basic systems and 150 data points with ID systems. To improve the network convergence, all the data were scaled to $[-1, 1]$ according to the extreme values of the training set.

3.3.2 Interaction network

For the second approach of predicting the interactions in the complex dislocation system, an interaction network (IN) was used. The IN implementation followed directly the network presented in [6]. The idea of the IN model is first to learn how physical objects of the system interact, and then to learn the relation between the interactions and the dynamics of the objects. Thus, the IN provides two applicable features for physics research: First, it can learn the underlying interactions and this way extract a physical model from the system. Second, it can predict the development of the system with the acquired dynamics. In the case of the dislocation system of the simulations, the dynamics simply reduce to the velocities of the dislocations. Learning the interaction from quasistatic simulations proved to be more challenging, so the IN was trained with data from simulations with constant external stress instead.

The input of the IN consists of a full description of the system which in this case consists of the dislocation positions x and y, Burgers vector sign s and a variable indicating whether the dislocation is annihilated. These are collected to matrix $\mathbf{O}$ which is of size $D_S \times N_O$ where D_S is the number of state variables (here this is four) and N_O is the number of dislocations. The state matrix is fed to IN with matrix $\mathbf{X}$ representing external effects and two permutation matrices, $\mathbf{R}_R$ and $\mathbf{R}_S$. $\mathbf{X}$ is a $1 \times N_O$ matrix, comprising of the external stress acting on each dislocation, while the permutation matrices are both $N_O \times N_R$, where N_R denotes the number of relations. As all the dislocations are interacting with each other, $N_R = N_O(N_O - 1)$. The subscripts R and S refer to receiver and sender of the interaction, respectively. $\mathbf{R}_R$ and $\mathbf{R}_S$ are constructed so that every dislocation pair is taken into account exactly once.

Step-by-step guide of the IN operating principles is presented in Figure 15. The first step in IN is to calculate matrix multiplications $\mathbf{OR}_R$ and $\mathbf{OR}_S$, as these represent the state of every receiver and sender of the interactions. Then, these are concatenated to form a $2D_S \times N_R$ matrix $\mathbf{B}$ (considering the interaction term, it is evident that the sign of the receiver is irrelevant and erasing column of the receiving dislocation sign from $\mathbf{B}$ slightly improves IN performance). $\mathbf{B}$ acts as input to the first MLNN of the IN, f_a. The output from f_a is a matrix $\mathbf{E}$ of size $D_E \times N_R$, where the N_R interactions are individually represented with D_E parameters. Afterwards, $\mathbf{E}$ is multiplied with $\mathbf{R}_R^T$ to form $D_E \times N_O$ matrix $\bar{\mathbf{E}}$ which contains the total interactions acting on a dislocation. Finally, matrices $\mathbf{O}$, $\mathbf{X}$ and $\bar{\mathbf{E}}$ are concatenated to a $(Ds + 1 + D_E) \times N_O$ matrix $\mathbf{C}$ (again for the purposes of this thesis, $\mathbf{O}$ includes unessential information concerning the velocity, as the absolute position

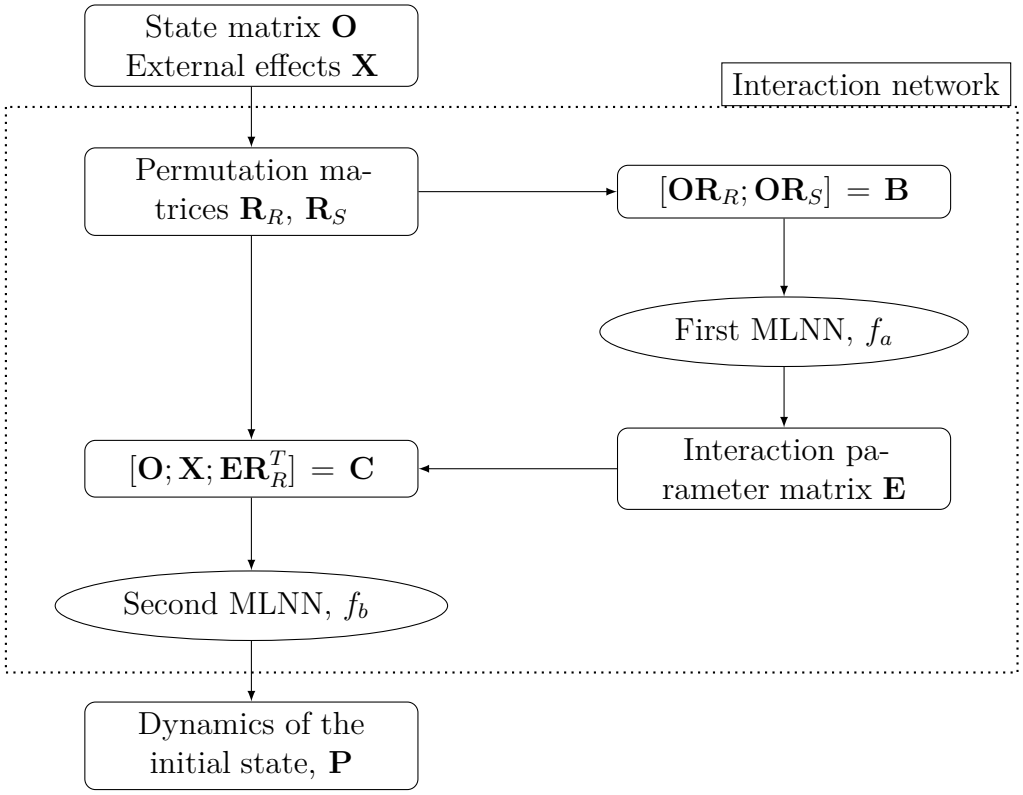


Figure 15: Flowchart of the IN.

of the dislocation does not affect the velocity and, thus, this is discarded). This is fed into the second MLNN, f_b, which outputs vector $\mathbf{P}$ representing the velocity of every dislocation.

Training the IN proved to be substantially complex. The approach of training with systems of few objects and then scaling upwards, which was taken in [6], was here impossible due to the poor variance in structures of few dislocations. In the small systems, the dislocations avoided configurations of close neighbours. Moreover, scaling the same IN to larger systems was practically impossible, considering that the simulated dislocation interactions of Equation (9) are dependent on the system size L. Hence, the training was conducted with systems initialized with 100 dislocations. For the training set, 225 initialized dislocation systems were simulated with constant stress ranging from 0.11 to 0.17, and from these simulations, 50000 images were chosen randomly. The test set was gathered from 25 simulations with the corresponding stresses. Due to the large number of dislocations per image, the training was time-consuming. Therefore, proper optimization of the MLNNs f_a and f_b was omitted as only a few architectures were tested. For the final implementation, both MLNNs consisted of two hidden layers, first with 100 and second with 50 neurons. Additionally, the number of parameters for interaction representation was set to $D_E = 10$.

4 Results

Here, the analysis of the simulation results and the actual predicting are presented. Firstly, aspects concerning the choice of features to describe the dislocation systems are discussed in Section 4.1. Secondly, the predictions of the system response to the loading using the chosen features are shown in Section 4.2. Thirdly, Section 4.3 explains characteristics of the predictions. And finally, Section 4.4 presents the performance of the interaction network applied to the dislocation simulations.

4.1 Feature extraction

To be able to do any predictions for the 2D dislocation system, the first step was to find a proper description of the initial configuration for the input of the MLNN – using the straightforward approach of dislocation positions would have been senseless, as the absolute positions have little meaning due to the periodic boundaries. Moreover, relative positions of the dislocations contain no information of the complex interactions and dynamics that the dislocations experience. Thus, the feature extraction required some creativity to obtain the best possible input parameters for the network.

A natural starting point was the stress fields σ_{sf} of the initial dislocation systems. The stress field of the previously considered configuration is shown in Figure 16. As the stress fields differ from system to system, these could partly explain the differences in the response for the external stress. For the input, statistics of the fields were calculated. These statistical features included the sum and variance of both the field σ_{sf} and the absolute field $|\sigma_{\mathrm{sf}}|$, and additionally the kurtosis and skewness of σ_{sf}. Also the median of $|\sigma_{\mathrm{sf}}|$ was used. Although this correlated with the mean absolute field, it still contained some new information as they improved the fitting results. Considering the interpretation of the statistical parameters, for instance the sum of the field measures if the dislocations lie in a balanced arrangement causing local field to vary evenly (sum near 0), or if the dislocations form local, irregular structures that produce regions of constant stress (sum differs from 0). In a similar sense, the variance, kurtosis and skewness measure the level of irregularity of the structures.

Another approach was to divide the internal structures to the ones that are stable and to the ones that are active. Hence, the field of the geometrically necessary dislocation (GND) density ρ_{GND} was calculated. In the simulated 2D system, local GND density is obtained by computing the difference of the local densities of dislocations with positive and negative Burgers vector [19],

$$\rho_{\mathrm{GND}} = \rho_{+} - \rho_{-}. \tag{24}$$

This way, in the regions where the structure is stable, that is no dislocations or any number of dislocation dipoles, ρ_{GND} is zero, while uneven structures yield

differing ρ_{GND}. The computation was carried in two ways. First, for the more crude description of the dislocation balance, the system was halved to two and the GND density difference between the halves was calculated. The division was done both in parallel and perpendicular to the glide plane. For the finer details, the system was divided to a 50×50 grid and ρ_{GND} was calculated in every cell. This is demonstrated with Figure 17 where a sparser grid (10×10) and the corresponding GND density field are shown for the previously seen system.

Again, representation of the GND density had to take PBC into account. Therefore, origin of the grid division was shifted in the x- and y-directions to obtain the grid with the largest deviations in ρ_{GND}. Additionally for the finer division, the Fourier transformation of the field was calculated. This enabled the description of the field $\rho_{\mathrm{GND}}(m, n)$ with the Fourier coefficients

$$f_{kl} = \sum_{m=0}^{49} \sum_{n=0}^{49} \rho_{\mathrm{GND}}(m, n) \exp\left[-2\pi i \left(\frac{mk}{50} + \frac{ln}{50}\right)\right], \tag{25}$$

which neglect the absolute positions of the cells and focus on the periodic field structure. Interestingly, the cross terms ($k \geq 1$ and $l \geq 1$) and higher terms ($k \geq 4$ or $l \geq 4$) of the Fourier transform showed no improvement on the predictions and, hence, they were discarded from the input. The six remaining coefficients are here denoted by f_{xi} and f_{yi}, where $l = 0$ in the first pair, $k = 0$ in the latter pair, and i is the coefficient index. Hence, the finer description of ρ_{GND} is basically the same as the bisection of the system, only difference being the finer scale.

As the third and final approach for describing the initial structure, the idea was

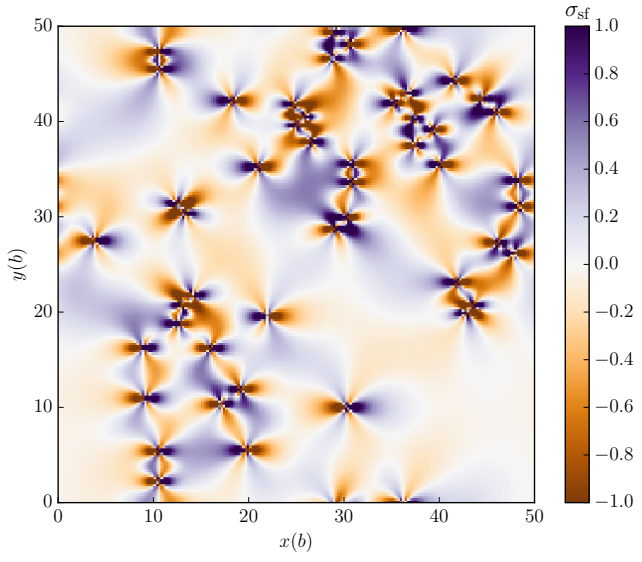
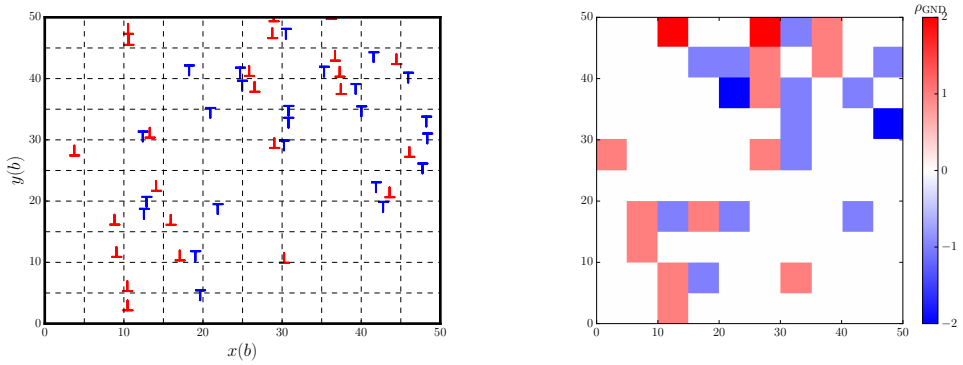


Figure 16: Stress field σ_{sf} of the dislocation configuration seen in Figure 7. The values near the dislocations are bounded as the forces diverge there, and the field is scaled with the bounding value to $[-1, 1]$.



(a) (b)

Figure 17: (a) The cells where to calculate ρ_{GND}. (b) The corresponding ρ_{GND} field.

Table 1: Input parameters for the regression network

sum of σ_{sf}	sum of $\lvert\sigma_{\mathrm{sf}}\rvert$
variance of σ_{sf}	variance of $\lvert\sigma_{\mathrm{sf}}\rvert$
skewness of σ_{sf}	median of $\lvert\sigma_{\mathrm{sf}}\rvert$
kurtosis of σ_{sf}	number of dislocations
f_{x1} of ρ_{GND} field	f_{y1} of ρ_{GND} field
f_{x2} of ρ_{GND} field	f_{y2} of ρ_{GND} field
f_{x3} of ρ_{GND} field	f_{y3} of ρ_{GND} field
ρ_{GND} difference in x-direction	ρ_{GND} difference in y-direction
dislocation wall count	maximum wall height
average wall height	(stress from ID, σ_{ID})

to count the number of formed dislocation walls. Figure 11 already showed that the dislocations tend to form walls where the dislocations of same sign lie more or less on the same vertical line. These walls move as a joined structures and, therefore, have effect on the system response to the external loading. Walls of different sizes, starting from a few dislocations, were seen in all of the initial systems. To identify dislocation that belong to the same wall, thresholds for distances in x- and y-directions were defined as $3b$ and $10b$, respectively. Other thresholds were also tested, but that did not affect the results significantly.

To summarize, all the regression network input parameters are presented in Table 1. For ID system, also the stress in the end of the initialization procedure, σ_{ID} could be included as input. Thus, there were altogether 19 input parameters plus one for ID systems.

4.2 Predicting the stress–strain -response with the regression network

The results of the training are presented in Figure 18, where the score of the network predicted stress $\sigma_{\text{prediction}}$ for the different data sets are plotted as a function of strain. Scores obtained for networks with different loading routines and input features –

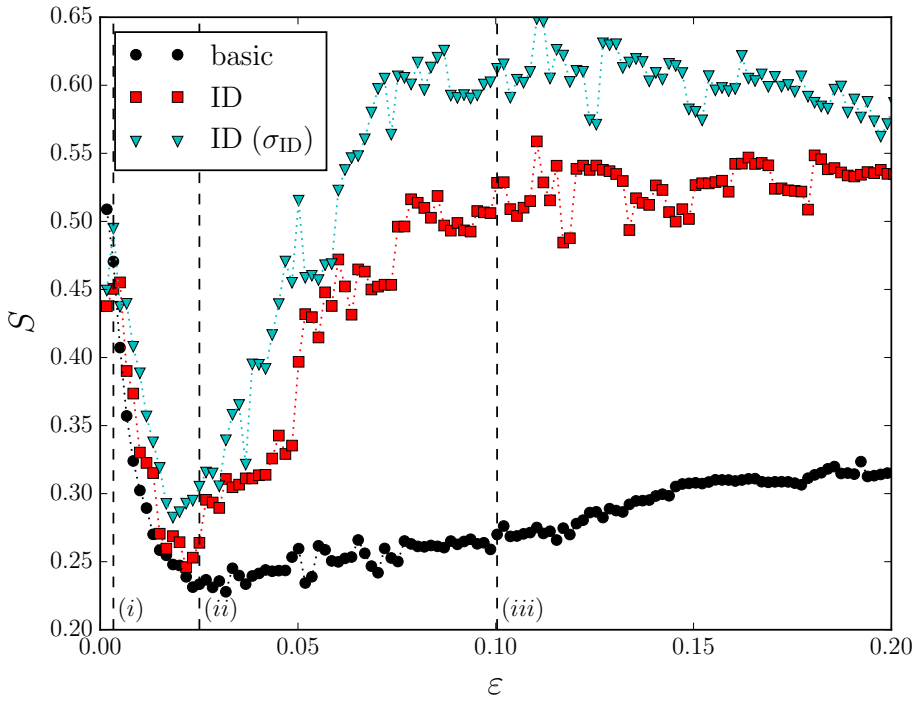


Figure 18: Score of the predicted stress values as a function of strain. Black circles denote the scores for the data set driven with the basic routine, while the red squares and turqoise triangles show the scores of ID system stress predictions for networks with and without the stress value input from ID, respectively. If the stress values are predicted also for even smaller values of strain than seen in the figure, the score does not approach unity but it remains close to 0.5.

basic, ID and ID with additional input of σ_{ID} – are denoted with different symbols. In general, scores remain well below the maximum value $S = 1$, which implies that the predictions are far from perfect. Also, inclusion of σ_{ID} improves the score as the networks trained without it perform worse at every strain than the networks trained with it.

Plots of the predicted stress versus the actual stress are shown in Figures 19-21 for the different sets at the strain values $(i) - (iii)$ indicated in Figure 18. For all of the three sets the scores are near 0.5 for small strains (i) in the start of the stress ramp. Of course, the relatively successful predictions of the small strain could

be anticipated, as the systems remain close to the initial configuration, which is described by the input parameters.

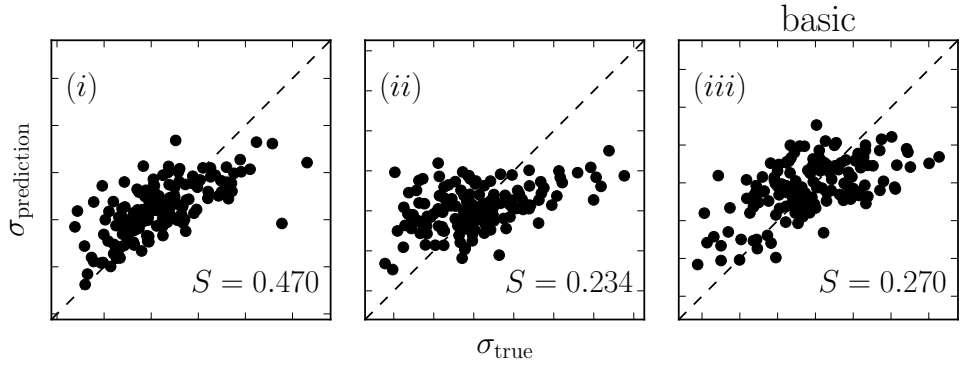


Figure 19: Predicted stress $\sigma_{\text{prediction}}$ versus the simulation stress value σ_{true} for the three strains $(i) - (iii)$ marked in the previous figure. The used data set was obtained with the basic loading routine. The dashed line shows $\sigma_{\text{prediction}} = \sigma_{\text{true}}$.

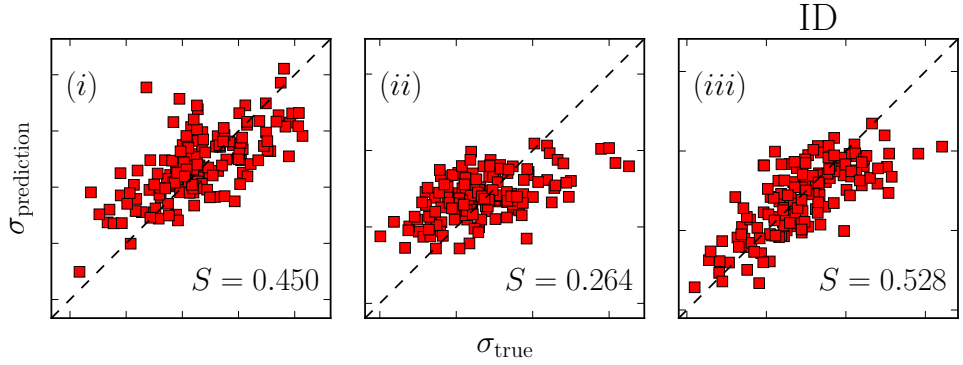


Figure 20: Predictions from networks for the ID test data.

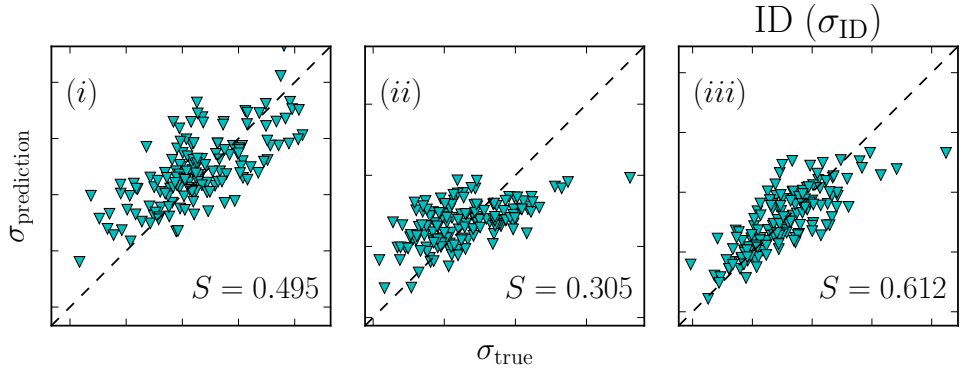


Figure 21: Predictions from networks for the ID test data, now with the addition of ID stress to the input.

However, the predictability of the stress decreases quickly. Around strain (ii), the scores of all sets have dipped. Thus, the prediction figures $\sigma_{\text{prediction}}$ show bad performance of the networks. Not much correlation can be seen between the predictions and the actual values as the predictions lie in a slightly tilted blob. Nevertheless, the score of the predictions increases after the large dip. For the systems

from basic simulations, the growth is slow and, hence, the plotted predictions at strain (*iii*) have improved only a little compared to (*ii*). Conversely, the predictability of the stress in the ID systems improves drastically with larger strains. Scores towards the larger strains actually beat the score of the smallest strain which is somewhat surprising. Indeed, predicted stress values at (*iii*) show stronger similarity with the simulation values from ID systems, as demonstrated by Figures 20 and 21.

4.3 Performance of the regression network

All in all, the regression networks succeeded only partially to learn the stress-strain response. The relatively good performance for small strain in the start of the simulation is apparent, as the systems have not changed much from the initial configuration which is described by the input of the model. The rapid decrease of the scores becomes evident with the observation of the average avalanche activity as a function of the strain. Indeed, the dislocation avalanches complicate the predicting.

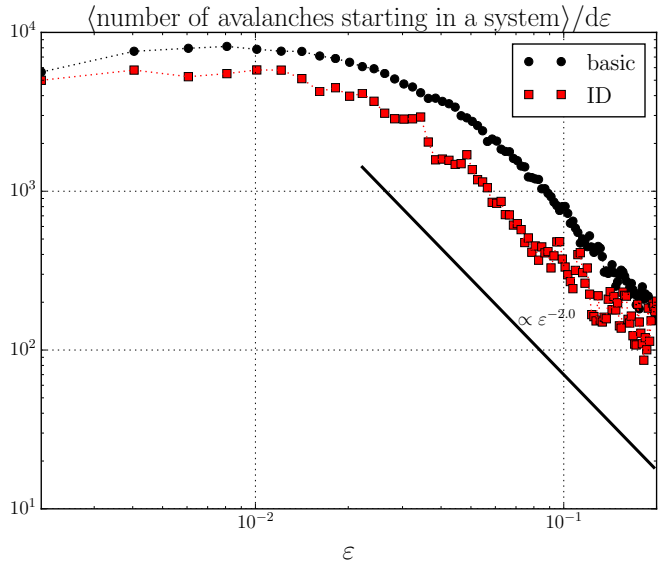


Figure 22: Rate of avalanche starts per system. In ID systems, the number of avalanches is smaller thoughout the simulations.

In Figure 22 the average avalanche starting rate, or in other words the average number of avalanches starting with strain, is plotted. This shows that the number of starting avalanches is highest at the smallest strains, while after the initial active strains the number decays as $\sim \varepsilon^{-2.0}$. As hinted by Figure 8, the avalanches with small strains are on average smaller than with large strains. Also Figure 23 confirms this. It shows the sum of sizes of the starting avalanches per system. (Again, the sum is scaled so that it represents the avalanche size per increasing strain.) In the start of the simulation, the sum of the avalanche sizes is small despite the large number of avalanches.

After the start of the simulations, the avalanche sum begins to increase towards a maximum. For ID systems, the maximum is larger and becomes with smaller strain than for the basic systems. After the maximum, the sum starts to decrease slowly, though for ID systems the sum first drops suddenly. This depicts the phase of the simulation where many of the systems are undergoing large avalanches and in these systems no new avalanches are starting. Thus, the avalanche activity appears to cause the predictability decrease of the stress values as the increasing activity coincides with the decline of the prediction scores. And similarly, the decrease of the avalanche activity occurs together with the following score increase. In a sense, the predictability of the stress values arises from the stress-strain curve itself as the parts of the curve with less avalanches are easier to predict.

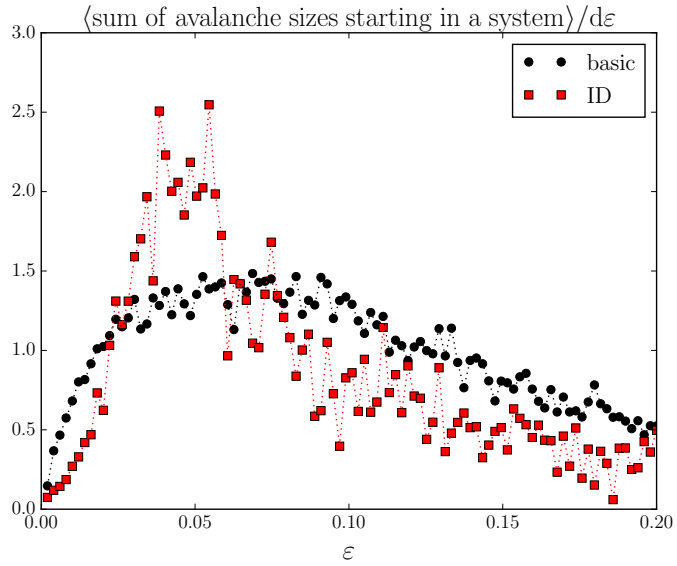
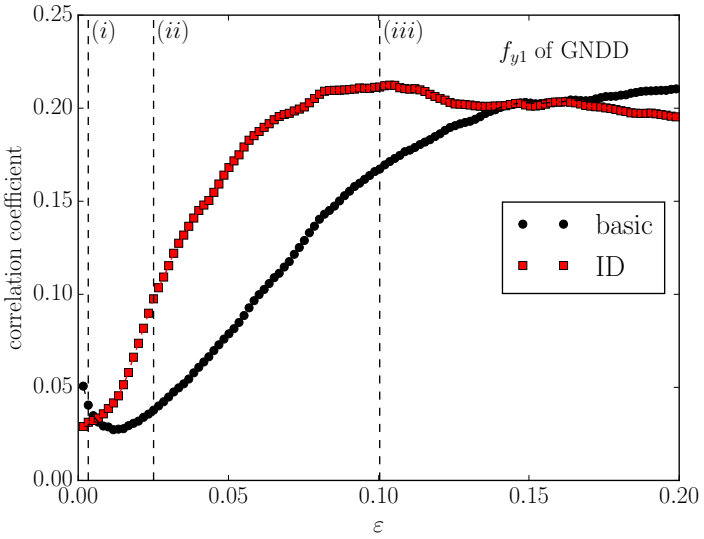


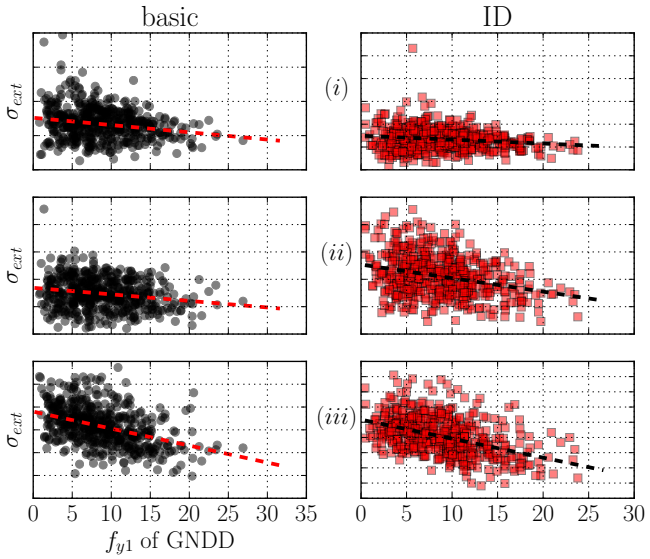
Figure 23: Sum of sizes of the starting avalanches per increasing strain. The sum starts to decrease after its maximum as more and more systems have ongoing dislocation avalanches and no new avalanches are starting. The noise of the plot originates from the binning of the strain values used for analysis, as single large avalanches cause the deviation to the average taken over all systems.

Another feature of the scores was the difference between the systems from basic and ID schemes. Predictability of the ID systems recovered so that the score was better with large strain predictions than with small strain. For basic systems, the score as well increased but much more slowly. Obviously this originates from the formed structure during the initial deformation, which was indicated by the dislocation-dislocation correlation functions plotted in Figure 9. Due to the initial structure, the ID systems are on average stronger and also the amount of avalanches is smaller as illustrated by Figures 10 and 22, respectively.

For the predictability of the stress in the latter stages of the simulation, the initial structures are crucial. This can be confirmed by closer observation of the



(a)

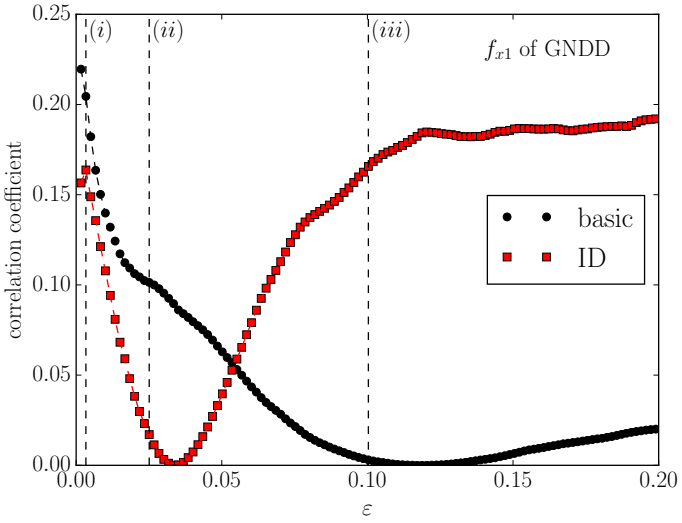


(b)

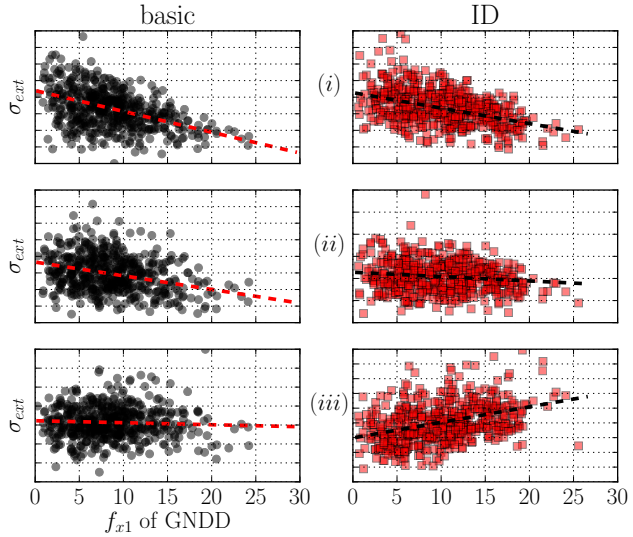
Figure 24: (a) Correlation coefficient of linear fit between f_{y1} and σ_{ext} and (b) the data with the linear fits for the three strains $(i) - (iii)$. Although the linear fit is not a perfect representation of the data, it shows the presence of some dependency.

Fourier coefficients f_{x1} and f_{y1} that depict the imbalance of GND density in the system parallel and perpendicular to the glide plane, respectively. Figure 24 shows the correlation coefficient of linear fit between f_{y1} and σ_{ext} as a function of the strain, along with the corresponding point clouds for the three chosen strains. f_{y1} is in a sense a permanent descriptor of the system. Due to the dislocation movement restriction to glide plane, f_{y1} remains same through the simulation. Plot of the

correlation coefficients shows that the imbalance of the GND density has an increasing influence on the flow stress during the simulation: the larger the coefficient, the smaller the external stress with larger strains. In other words, larger fluctuation of the GND density in y-direction tends to yield weaker systems because the correlation between f_{y1} and σ_{ext} is negative, and this holds for systems initialized with both basic and ID schemes.



(a)



(b)

Figure 25: (a) Correlation coefficient of linear fit between f_{x1} and σ_{ext} and (b) the data with the linear fits for the three strains $(i) - (iii)$.

On the other hand, f_{x1} describes the dislocation imbalance in the direction of the glide plane. Correspondingly, Figure 25 shows the correlation coefficient for f_{x1}

and σ_{ext}. The influence of f_{x1} on the stress values is larger in the beginning of the simulation than the influence of f_{y1}. However, accumulating more strain causes the correlation disappear – faster for the ID systems than for the basic systems – but the correlation of f_{x1} and σ_{ext} reappears for the ID systems, while for the basic systems the correlation is basically lost. This originates from the wall structure of the initial ID configurations that especially f_{x1} captures: As the dislocations tend to form walls, the pre-existing walls in ID systems and the dislocation structure after the stress ramp are similar. The initial basic configurations have less walls ready and, thus, the structure in the end of the ramp is different from the initial structure. Lastly, the stress dependence on f_{x1} shows also an intriguing feature, as the correlation is reversed. In the beginning, larger GND density differences cause the system become weaker while after the reappearance of the correlation for ID systems, the effect is the opposite. To explain this, a closer study of dislocation structures and motion in individual simulations is required.

4.4 Predicting individual dislocation velocities with the interaction network

Comparing to the stress-strain predictions of the regression network, IN managed to learn the dislocation interactions far better. Figure 26 shows the predictions versus the actual single dislocation velocities in nine test systems, $(a) - (i)$. The velocities were calculated for a random set of images from the simulation and the images were treated as separate cases. Now, the score S settled near unity as the actual values coincided well with the network output. Additionally, the score appeared to be higher with larger σ_{ext}.

Starting from the initial dislocation positions, IN could also generate dynamic predictions. In practice, this was easy to implement with a time-integration routine that updated the dislocations with the velocities from IN and revised the possible dislocation annihilations after every step. In Figure 27, the average velocity v_{average} of the dislocations is plotted for the test systems $(a) - (i)$ from the simulation and the dynamic prediction. Although v_{average} is not an absolute measure of the coinciding dynamics, it is used here as comparing for instance individual dislocation positions or velocities at every time step would be arbitrary due to the possible discrepancies in existing dislocations. The figure shows that generally the predictions coincide with the simulations, especially in the start of the loading. Most predictions seem to capture the simulation dynamics as the velocity curve shapes are similar, but slightly shifted. In these systems, the dislocations indeed reach similar final configurations when compared separately. But in some systems, for instance (a) and (g), the predictions fail towards the end of the loading. Naturally, this is due to the small errors in the velocity predictions, that accumulate to push the system into a totally different configuration.

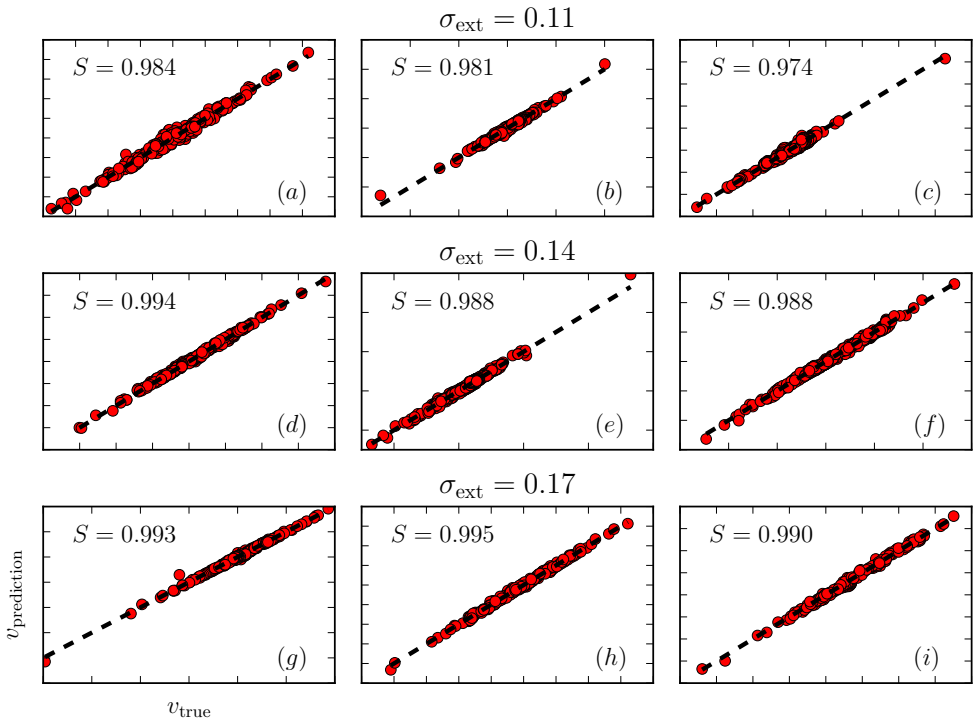


Figure 26: IN predictions plotted as a function of the actual velocities of the single dislocations in test systems $(a)-(i)$. For systems $(a)-(c)$, $(d)-(f)$ and $(g)-(i)$ the applied external stress was set to $\sigma_{\text{ext}} = 0.11$, $\sigma_{\text{ext}} = 0.14$ and $\sigma_{\text{ext}} = 0.17$, respectively. The dashed lines represent $v_{\text{prediction}} = v_{\text{true}}$.

To obtain an overview of the interaction IN learned, velocity field output of IN was computed near a test dislocation and compared with the exact field of Equation (10). The test dislocation was assigned with a positive Burgers vector and it was placed to the origin. Figure 28 illustrates the exact and IN velocity fields along with the absolute and relative error between the fields for a dislocation with negative Burgers vector. A glance at the fields suggests that the IN has learned the interaction particularly well, when the dislocations are separated by some distance, while the learned field near the test dislocation is not as strong as the exact field. This is confirmed by the error fields. A simple explanation would be that the training set does not include many images with different signed dislocations extremely close to each other on almost same glide plane. This is due to the fact that the approaching dislocations experience the largest velocities that lead them to annihilate.

Interestingly, the absolute error is largest left from the test dislocation, while the field above, right and below the test dislocation has been acquired by the IN. Indeed, closer inspection of the error figures indicates that the largest errors occur in the regions where the velocity is reversed to the external field, that is where the negative dislocation travels to the positive x-direction. On the other side of the test dislocation, the IN has had no such problems. Again, this is most likely arising

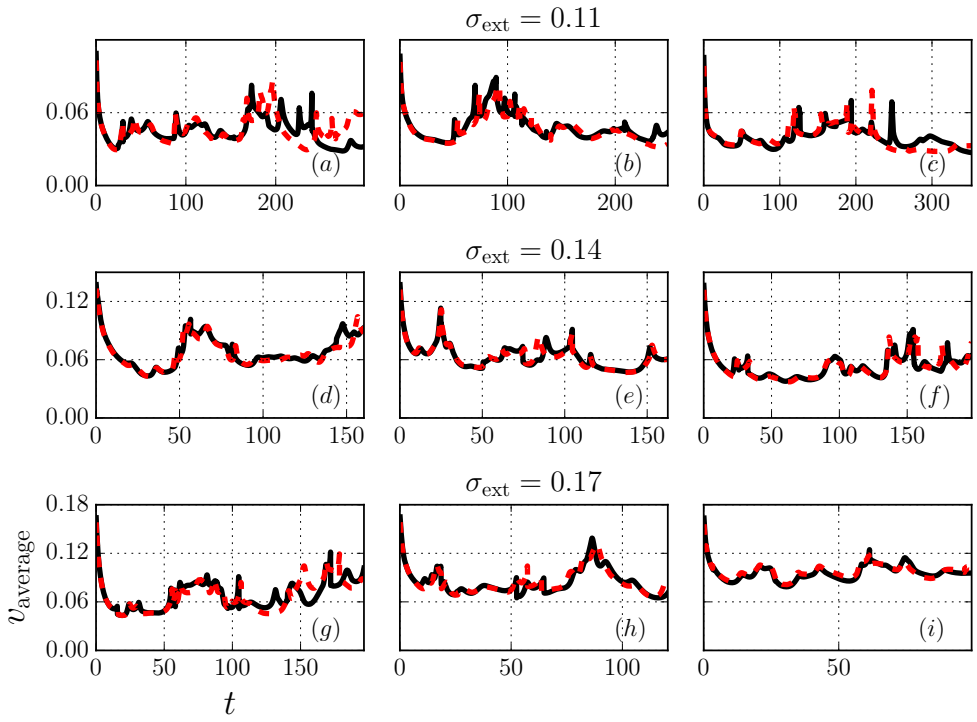


Figure 27: v_{average} plotted as a function of time t from the dynamic predictions (dashed red line) produced by the IN for the test systems $(a) - (i)$ compared with the simulation results (solid black line). Simulations were computed until the strain in system reached $\varepsilon = 0.2$.

from the training set, where dislocations travelling to the 'wrong' direction form a minority. Thus for the IN, learning the field on the other side of the test dislocation seems to be easier as the dislocation velocities are there to the 'assumed' direction.

Correspondingly, Figure 29 shows the velocity fields and their difference for dislocations with positive Burgers vector. The relative error is again more significant for regions where the velocity should be against the external field. The effect is emphasized with the same signed dislocations, as they strongly repel each other and seldom form close configurations. Therefore, in the case where the dislocations reach these rare configurations during creep, the IN starts to falter and change from the actual simulation. In addition to larger training set and better network architecture, another possible way to improve the IN performance could be to use even larger dislocation systems, because there the possible dislocation configurations are more diverse.

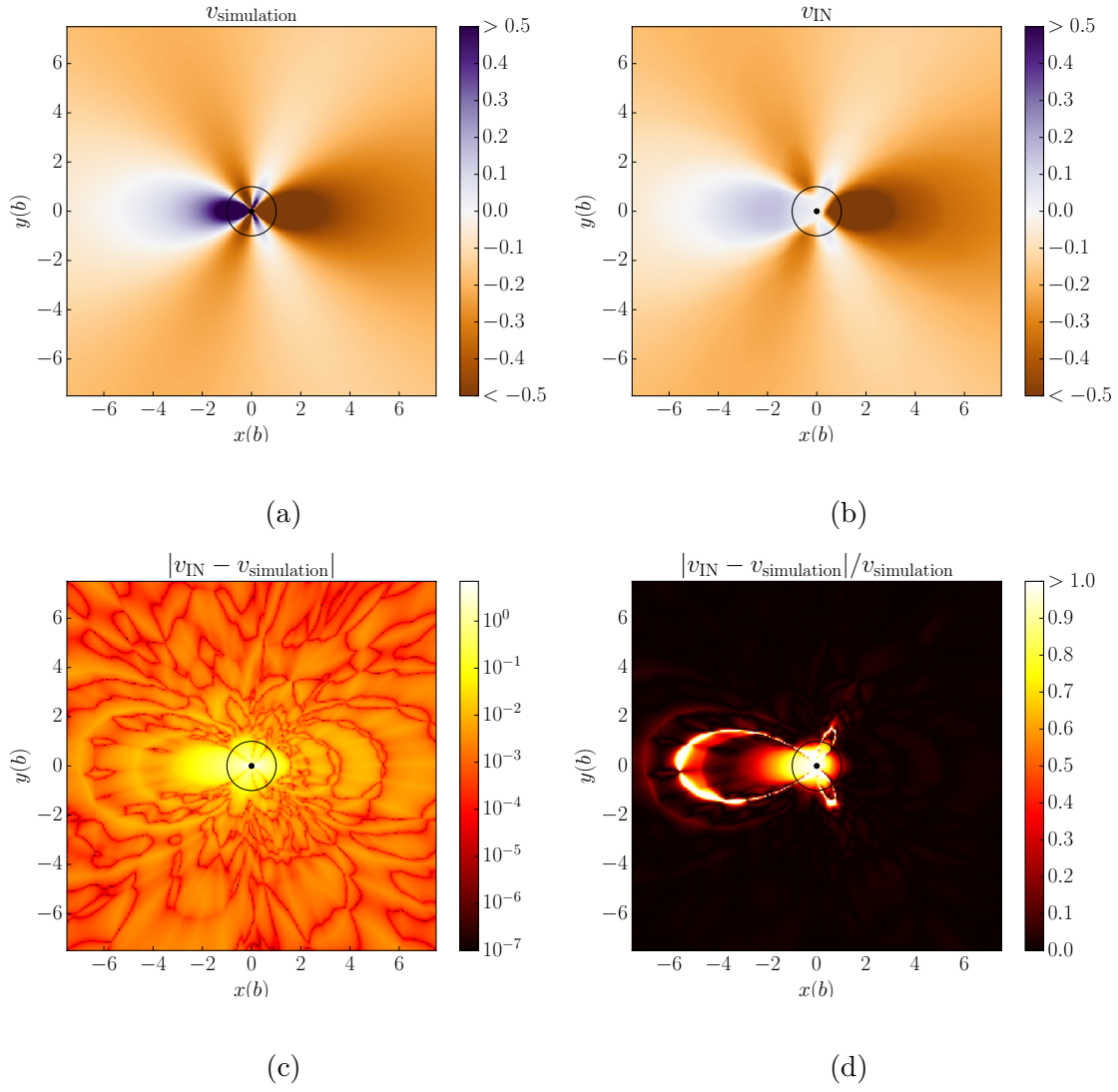


Figure 28: (a) Exact velocity field of Equation (10), (b) IN prediction of the field, (c) absolute difference and (d) relative difference between the fields. The test dislocation, marked with the black dot, was assigned with a positive Burgers vector, while the field is for a dislocation with negative Burgers vector. The black circle shows the core of the test dislocation, where the dislocations would annihilate. External stress was set to $\sigma_{\text{ext}} = 0.17$.

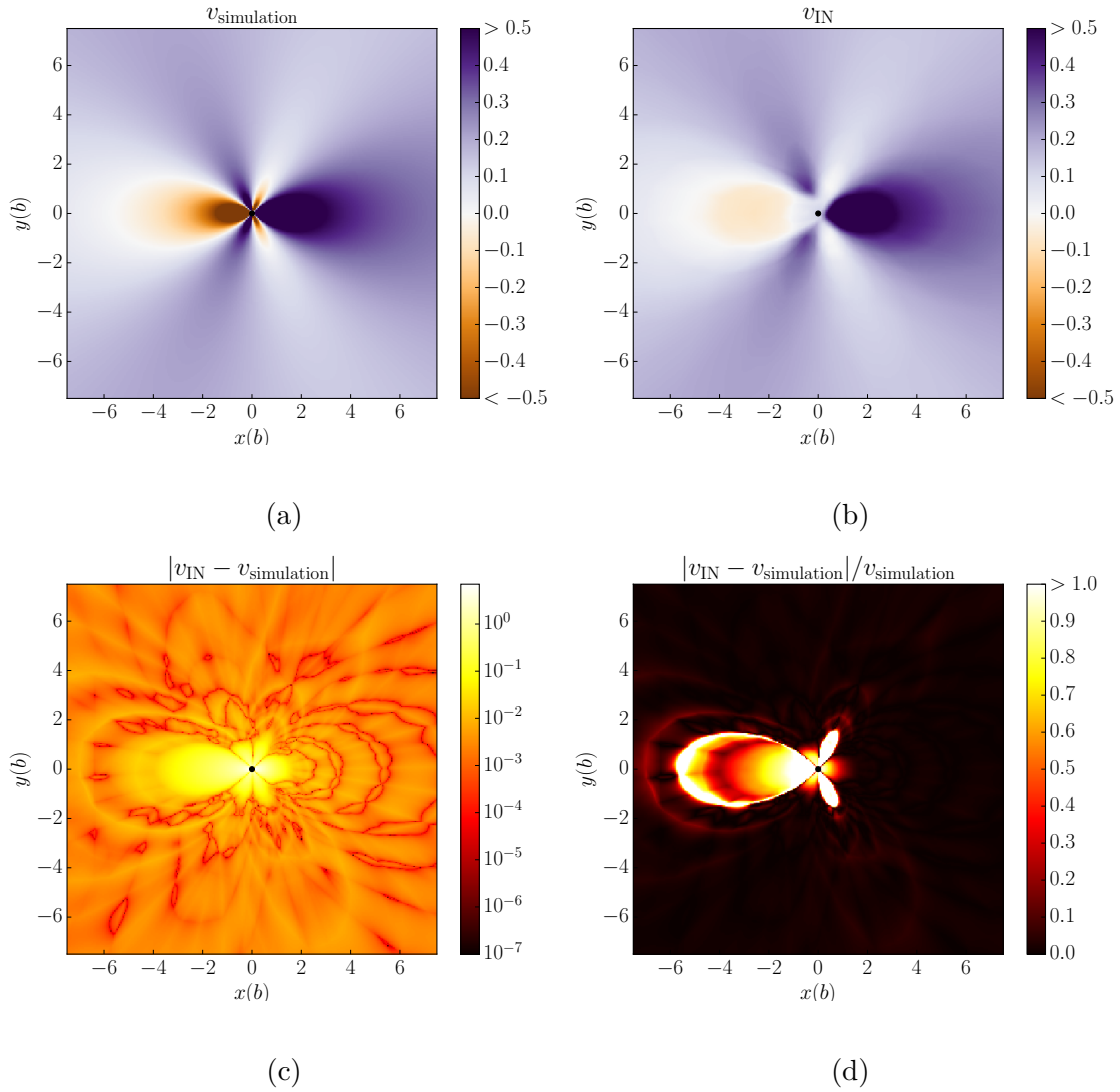


Figure 29: (a) Exact velocity field of Equation (10), (b) IN prediction of the field, (c) absolute difference and (d) relative difference between the fields. Now, both of the dislocations have positive Burgers vector and $\sigma_{\text{ext}} = 0.17$.

5 Discussions of future work

The predictions of the stress values of the 2D DDD simulations with the quasistatic stress ramp showed that, although the strain bursts seem to prevent absolute success, there is some connection between the stress response and the initial dislocation configuration of the system. Hence, the evident next step is to study if same kind of predictability can be recovered from more complex systems, namely 3D DDD simulations. However, this contains a new set of challenges.

There are several implementations for corresponding 3D DDD simulations. An open source software called ParaDiS (Parallel Dislocation Simulator) has been developed at Lawrence Livermore National Laboratory [38], and it has also been modified to include quasistatic loading by Lehtinen et al. [14]. An example stress-strain response of a simulated system with ParaDiS is presented in Figure 30. The idea in ParaDiS is to discretize the dislocation line to segments. Then, the equations of motion are computed for the discretization nodes where the segment-wise interactions between the dislocations are obtained by line-integration of the Peach-Koehler force. The code also acknowledges the presence of short-range interactions near the dislocation cores and long-range interactions. Thus, the 3D DDD simulations take notably longer times to compute than their 2D counterparts, which is also a problem for the predicting: obtaining large datasets for training is computationally expensive. Although the simulations for [14] are already available for analysis, the number of simulations is measured in hundreds instead of thousands.

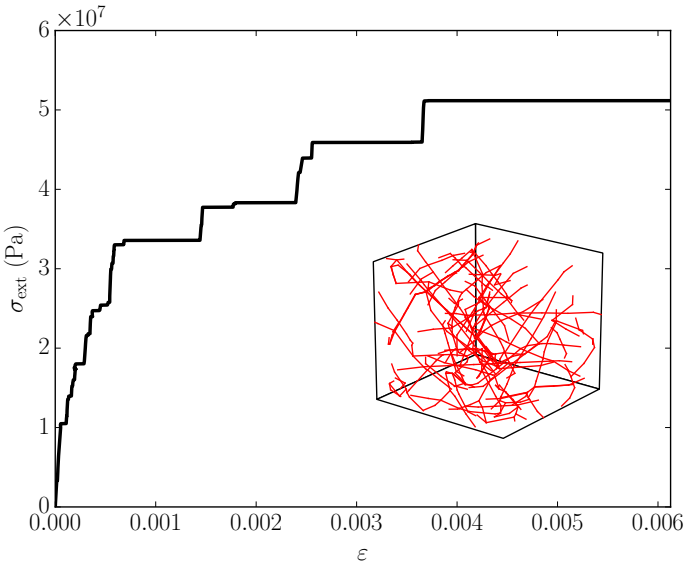


Figure 30: Example stress-strain curve of a 3D DDD simulation of face-centered cubic crystal with material parameters of aluminium conducted in [14]. The inset shows the initial relaxed simulation box of size $L = 1.43\,\mu\text{m}$ with the dislocation lines before the stress ramp.

Another challenge arises from the characterization of the initial system. In 3D, most of the measurable parameters, for instance the stress field and the GND density, are tensors. Therefore, the amount of possible descriptors is in a sense much larger than for the 2D system. Moreover combining this to the fact that obtaining a large training set is computationally inconceivable, finding a reasonable set of descriptors becomes essential. This is because using all the retrieved information of the initial system would yield far too many input parameters compared to the number of samples.

On the other hand, applying interaction networks on dislocation dynamics to predict material properties is a bit challenging. For instance, training the interaction network to learn the dynamics of dislocations in real life materials would require ongoing measurements of single dislocations during loading experiment which is unattainable with current measuring devices. Nevertheless, IN showed the capability of learning the interactions of a complex system with dozens of objects. This could be utilized in experimental problems where the elementary interactions are unknown but the dynamics are observable. Another option would be to use IN as an alternative for the simulations in the sense that the IN would learn the dynamics from ready simulations and then perform the entire computation procedure instead of calculating any new simulations. For this to be reasonable, the computing time should be shorter, which requires that the simulation problem includes heavy computing of a system with a sensible number of objects. Considering the 2D DDD simulations, IN trained with an optimized procedure and dataset would not hasten the simulation due to the relatively easy computations of the model.

6 Summary

To conclude, results of this thesis managed to show that in 2D DDD simulations, there is a connection between the stress-strain response to quasistatic stress ramp and the initial dislocation configuration. The initial dislocation structure was described with statistical parameters of the stress field, Fourier coefficients of the GND density field and straightforward calculation of pre-existing dislocation walls. The descriptors were input to a regression neural network that then predicted the stress for different strain values.

The predictions of the stress values were not perfect but correlation between the actual and predicted values was apparent. Predictability of the stress was naturally best at the start of the simulation for the systems initialized with the basic scheme. On the other hand for initially deformed systems, the score of prediction was better towards the end of the simulation due to the dislocation structures formed during the initial deformation. Additionally, information from the initial deformation, namely the stress in the end of the deformation, improved the predictions of the ID systems. The bursty behaviour of the system, that is the dislocation avalanches, appeared to complicate the predicting. This was hinted by the fact that the scores of the predictions experienced a decline that partly coincided with the maximum of starting avalanche activity. The next step is to study if 3D DDD systems contain similar predictability of the stress response, as this would be a stronger proof of possibility to predict material properties from the initial dislocation configurations.

In this thesis, the application of an interaction network, that first learns the interactions inside the system and then their influence on the dynamics, was also tested on the 2D DDD simulations. The interaction network learned the underlying interactions almost completely when the system was loaded with constant external stress. The only problem appeared to be the dislocations which were travelling against the external field due to strong interactions caused by close neighbours. For the used system, these events formed a small minority, so the inability of the interaction network to learn these could be overcome by better architecture of the network or better sampling of the training data. But in general, the interaction network showed capability to learn interactions from observations of dynamics of a complex system.

References

[1] Alava, M. J., Laurson, L., & Zapperi, S. (2014). Crackling noise in plasticity. *The European Physical Journal Special Topics* 223(11), 2353-2367.

[2] Weiss, J., Grasso, J. R., Miguel, M. C., Vespignani, A., & Zapperi, S. (2001). Complexity in dislocation dynamics: experiments. *Materials Science and Engineering: A* 309, 360-364.

[3] Ng, K. S., & Ngan, A. H. W. (2008). Stochastic nature of plasticity of aluminum micro-pillars. *Acta Materialia* 56(8), 1712-1720.

[4] Mueller, T., Kusne, A. G., & Ramprasad, R. (2016). Machine learning in materials science: recent progress and emerging applications. *Reviews in Computational Chemistry* 29, 186.

[5] Haykin, S. S. Neural networks: a comprehensive foundation. Tsinghua University Press, 2001.

[6] Battaglia, P., Pascanu, R., Lai, M., & Rezende, D. J. (2016). Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems* (pp. 4502-4510).

[7] Hirth, J. P., & Lothe, J. Theory of dislocations. Krieger Publishing Company, 1982.

[8] Hull, Derek, & David J. Bacon. Introduction to dislocations. Butterworth-Heinemann, 2001. Fourth edition.

[9] Bulatov, Vasily, & Wei Cai. Computer simulations of dislocations. Vol. 3. Oxford University Press on Demand, 2006

[10] Uchic, M. D., Dimiduk, D. M., Florando, J. N., & Nix, W. D. (2004). Sample dimensions influence strength and crystal plasticity. *Science* 305(5686), 986-989.

[11] Uchic, M. D., Shade, P. A., & Dimiduk, D. M. (2009). Plasticity of micrometer-scale single crystals in compression. *Annual Review of Materials Research* 39, 361-386.

[12] Ispánovity, P. D., Hegyi, Á., Groma, I., Györgyi, G., Ratter, K., & Weygand, D. (2013). Average yielding and weakest link statistics in micron-scale plasticity. *Acta Materialia* 61(16), 6234-6245.

[13] Szabó, P., Ispánovity, P. D., & Groma, I. (2015). Plastic strain is a mixture of avalanches and quasireversible deformations: Study of various sizes. *Physical Review B* 91(5), 054106.

[14] Lehtinen, A., Costantini, G., Alava, M. J., Zapperi, S., & Laurson, L. (2016). Glassy features of crystal plasticity. *Physical Review B* 94(6), 064101.

[15] Ovaska, M. (2016). Avalanches in plastic deformation of materials. Dissertation, Aalto University School of Science, Department of Applied Physics 2017

[16] Ispánovity, P. D., Groma, I., Györgyi, G., Csikor, F. F., & Weygand, D. (2010). Submicron plasticity: yield stress, dislocation avalanches, and velocity distribution. *Physical Review Letters* 105(8), 085503.

[17] Dimiduk, D. M., Woodward, C., LeSar, R., & Uchic, M. D. (2006). Scale-free intermittent flow in crystal plasticity. *Science* 312(5777), 1188-1190.

[18] Csikor, F. F., Motz, C., Weygand, D., Zaiser, M., & Zapperi, S. (2007). Dislocation avalanches, strain bursts, and the problem of plastic forming at the micrometer scale. *Science* 318(5848), 251-254.

[19] Ispánovity, P. D., Papanikolaou, S., & Groma, I. (2017). The Emergence and Role of Dipolar Dislocation Patterns in Discrete and Continuum Formulations. *arXiv preprint* arXiv:1708.03710.

[20] Carrasquilla, J., & Melko, R. G. (2017). Machine learning phases of matter. *Nature Physics* 13(5), 431-434.

[21] van Nieuwenburg, E. P., Liu, Y. H., & Huber, S. D. (2017). Learning phase transitions by confusion. *Nature Physics* 13(5), 435-439.

[22] Beach, M. J., Golubeva, A., & Melko, R. G. (2017). Machine learning vortices at the Kosterlitz-Thouless transition. *arXiv preprint* arXiv:1710.09842.

[23] Long, C. J., Hattrick-Simpers, J., Murakami, M., Srivastava, R. C., Takeuchi, I., Karen, V. L., & Li, X. (2007). Rapid structural mapping of ternary metallic alloy systems using the combinatorial approach and cluster analysis. *Review of Scientific Instruments* 78(7), 072217.

[24] Papanikolaou, S., Tzimas, M., Song, H., Reid, A. C., & Langer, S. A. (2017). Learning crystal plasticity using digital image correlation: Examples from discrete dislocation dynamics. *arXiv preprint* arXiv:1709.08225.

[25] Kong, L.X. & Hodgson, P.D. (1999). The application of constitutive and artificial neural network models to predict the hot strength of steels. *ISIJ International* 39(10), 991-998.

[26] Phaniraj, M. P., & Lahiri, A. K. (2003). The applicability of neural network model to predict flow stress for carbon steels. *Journal of Materials Processing Technology* 141(2), 219-227.

[27] Yassar, R. S., AbuOmar, O., Hansen, E., & Horstemeyer, M. F. (2010). On dislocation-based artificial neural network modeling of flow stress. *Materials & Design* 31(8), 3683-3689.

[28] Miguel, M. C., Laurson, L., & Alava, M. J. (2008). Material yielding and irreversible deformation mediated by dislocation motion. *The European Physical Journal B* 64(3-4), 443-450.

[29] Ovaska, M., Laurson, L., & Alava, M. J. (2015). Quenched pinning and collective dislocation dynamics. *Scientific Reports* 5, 10580.

[30] Van der Giessen, E., & Needleman, A. (1995). Discrete dislocation plasticity: a simple planar model. *Modelling and Simulation in Materials Science and Engineering* 3(5), 689.

[31] Zaiser, M., Miguel, M. C., & Groma, I. (2001). Statistical dynamics of dislocation systems: the influence of dislocation-dislocation correlations. *Physical Review B* 64(22), 224102.

[32] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65(6), 386.

[33] Funahashi, K. I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks* 2(3), 183-192.

[34] Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (pp. 315-323).

[35] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks* 61, 85-117.

[36] Alpaydin, E. Introduction to machine learning. MIT press, 2014.

[37] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint* arXiv:1603.04467.

[38] Arsenlis, A., Cai, W., Tang, M., Rhee, M., Oppelstrup, T., Hommes, G., ... & Bulatov, V. V. (2007). Enabling strain hardening simulations with dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering* 15(6), 553.